

PREFACE

The IBM Personal Computer Technical Reference Manual is designed to provide hardware design and interface information. This publication also provides Basic Input Output System (BIOS) information as well as programming support matter.

This manual is intended for programmers, engineers involved in hardware and software design, designers, and interested persons who have a need to know how the IBM Personal Computer is designed and works.

This manual has three sections:

Section-1

"HARDWARE OVERVIEW," features an overview of the system as a whole calling out specific items such as the System Unit, Keyboard, IBM Monochrome Display and the 80 CPS Graphics Printer.

Section-2

"HARDWARE," contains a description for each functional part of the system. This section also contains specifications for power, timing, and interface. Programming considerations are supported by coding tables, command codes and registers.

Section-3

"ROM and SYSTEM USAGE," describes BIOS as well as how to use BIOS, interrupt vector listings, memory map, vectors with special meanings, a cassette section, a keyboard encoding section, and a set of Low Memory Maps.

"APPENDICES," to address the ROM BIOS listing, an instruction set, logic diagrams, and expanded charts used to support specific hardware descriptions.



Technical Reference

CONTENTS

| | |
|--|-------------|
| SECTION 1. HARDWARE OVERVIEW | 1-1 |
| System Block Diagram | 1-6 |
| SECTION 2. HARDWARE | 2-1 |
| System Board | 2-3 |
| System Board Data Flow | 2-6 |
| I/O Channel | 2-8 |
| I/O Channel Diagram | 2-9 |
| System Board I/O Channel Description | 2-10 |
| System Board Component Diagram | 2-13 |
| Keyboard | 2-14 |
| Keyboard Interface Block Diagram | 2-15 |
| Keyboard Diagram | 2-16 |
| Keyboard Scan Codes | 2-23 |
| Keyboard Interface Connector Specifications | 2-24 |
| Cassette User Interface | 2-25 |
| Cassette Jumpers | 2-25 |
| Circuit Block Diagrams | 2-25 |
| Cassette Interface Connector Specifications | 2-27 |
| Speaker Interface | 2-28 |
| Speaker Drive System Block Diagram | 2-28 |
| I/O Address Map | 2-29 |
| System Memory Map | 2-31 |
| System Board and Memory Expansion Switch Settings | 2-34 |
| 5 1/4" Diskette Drives Switch Settings | 2-35 |
| Monitor Type Switch Settings | 2-35 |
| System Board Memory Switch Settings | 2-36 |
| 32, 64 and 64/256KB Memory Expansion Option Switch Settings | 2-38 |
| Power Supply | 2-43 |
| Power Supply Location | 2-44 |
| Input Requirements | 2-44 |
| DC Output | 2-44 |
| AC Output | 2-44 |
| Power Supply Connectors and Pin Assignments | 2-45 |
| Important Operating Characteristics | 2-46 |
| Over Voltage/Current Protection | 2-46 |
| Signal Requirements | 2-46 |

| | |
|--|-------|
| IBM Monochrome Display and Parallel Printer Adapter | 2-47 |
| Parallel Interface Description | 2-47 |
| IBM Monochrome Display Adapter Block Diagram | 2-48 |
| System Channel Interface | 2-49 |
| Lines Used | 2-49 |
| Loads | 2-49 |
| Special Timing | 2-49 |
| Data Rates | 2-49 |
| Interrupt and DMA Response Requirements | 2-49 |
| Modes Of Operation | 2-50 |
| Programming Considerations | 2-51 |
| Programming the 6845 CRT Controller | 2-51 |
| Sequence of Events | 2-51 |
| Memory Requirements | 2-51 |
| DMA Channels | 2-52 |
| Interrupt Levels | 2-52 |
| I/O Address and Bit Map | 2-52 |
| I/O Address Function | 2-52 |
| IBM Monochrome Display | 2-53 |
| Operating Characteristics | 2-53 |
| IBM Monochrome Direct Drive Interface and Pin Assignment | 2-54 |
| Color/Graphics Monitor Adapter | 2-55 |
| Color/Graphics Monitor Adapter Block Diagram | 2-57 |
| Major Components Definitions | 2-58 |
| Motorola 6845 CRT Controller | 2-58 |
| Mode Set and Status Registers | 2-58 |
| Display Buffer | 2-58 |
| Character Generator | 2-58 |
| Timing Generator | 2-58 |
| Composite Color Generator | 2-58 |
| Modes of Operation | 2-59 |
| Alphanumeric Mode | 2-59 |
| Color TV | 2-59 |
| Color Monitor | 2-60 |
| IBM Monochrome Display Adapter Vs. Color/Graphics Adapter Attribute Relationship | 2-61 |
| Color/Graphics Modes | 2-61 |
| Graphics Storage Map | 2-62 |
| Description of Basic Operations | 2-63 |
| Summary of Available Colors | 2-65 |
| Programming Considerations | 2-65 |
| Programming The 6845 Controller | 2-65 |
| 6845 Register Description | 2-66 |
| Programming the Mode Control and Status Register | 2-67 |
| Color Select Register | 2-67 |
| Mode Select Register | 2-68 |
| Mode Register Summary | 2-69 |
| Status Register | 2-69 |
| Sequence of Events | 2-70 |
| Memory Requirements | 2-70 |
| Interrupt Level | 2-70 |
| I/O Address and Bit Map | 2-71 |
| Color/Graphics Monitor Adapter Direct Drive, and Composite Interface Pin Assignment | 2-72 |
| Color/Graphics Monitor Adapter Auxiliary Video Connectors | 2-73 |
| Parallel Printer Adapter | 2-75 |
| Parallel Printer Block Diagram | 2-76 |
| Programming Considerations | 2-77 |
| Parallel Printer Adapter Interface Connector Specifications | 2-79 |
| IBM 80 CPS Graphics Printer | 2-80 |
| Printer Specifications | 2-81 |
| Setting the DIP Switches | 2-84 |
| Functions and Conditions of DIP Switch 1 | 2-84 |
| Functions and Conditions of DIP Switch 2 | 2-85 |
| Parallel Interface Description | 2-85 |
| Connector Pin Assignment and Descriptions of Interface Signals | 2-86 |
| Parallel Interface Timing Diagram | 2-89 |
| IBM Graphics Printer Modes | 2-90 |
| Printer Control Codes | 2-91 |
| Graphics Printer Character Sets | 2-102 |
| 5 1/4" Diskette Drive Adapter | 2-107 |
| 5 1/4" Diskette Drive Adapter Block Diagram | 2-108 |
| Functional Description | 2-109 |
| Digital Output Register | 2-109 |
| Floppy Disk Controller | 2-109 |
| Programming Considerations | 2-112 |
| Symbol Descriptions | 2-112 |
| Command Summary | 2-114 |
| Command Status Registers | 2-118 |

| | |
|---|-------|
| Programming Summary | 2-121 |
| DPC Registers | 2-121 |
| Drive Constants | 2-122 |
| Comments | 2-122 |
| System I/O Channel Interface | 2-122 |
| Drive A and B Interface | 2-124 |
| Adapter Outputs | 2-124 |
| Adapter Inputs | 2-125 |
| 5 1/4" Diskette Drive Adapter Internal Interface Specifications | 2-126 |
| 5 1/4" Diskette Drive Adapter External Interface Specifications | 2-127 |
| 5 1/4" Diskette Drive | 2-128 |
| Diskettes | 2-129 |
| Mechanical and Electrical Specifications | 2-130 |
| Memory Expansion Options | 2-131 |
| Operating Characteristics | 2-132 |
| Memory Module Description | 2-132 |
| Memory Module Pin Configuration | 2-133 |
| Switch Configurable Start Address | 2-133 |
| Game Control Adapter | 2-135 |
| Game Control Adapter Block Diagram | 2-135 |
| Functional Description | 2-136 |
| Address Decode | 2-136 |
| Data Buss Buffer/Driver | 2-136 |
| Trigger Buttons | 2-136 |
| Joystick Positions | 2-136 |
| I/O Channel Description | 2-137 |
| Interface Description | 2-137 |
| Joystick Schematic | 2-139 |
| Game Control Adapter (Analog Input) Connector Specifications | 2-140 |
| Asynchronous Communications Adapter | 2-141 |
| Asynchronous Communications Adapter Block Diagram | 2-142 |
| Modes of Operation | 2-143 |
| I/O Decode for Communications Adapter | 2-143 |
| Interrupts | 2-144 |
| Interface Description | 2-144 |
| Current Loop Interface | 2-145 |
| Voltage Interchange Information | 2-146 |

| | |
|--|-------|
| INS 8250 Functional Pin Description | 2-147 |
| Input Signals | 2-147 |
| Output Signals | 2-150 |
| Input/Output Signals | 2-151 |
| Programming Considerations | 2-151 |
| Asynchronous Communications Reset Functions | 2-151 |
| INS 8250 Accessible Registers | 2-152 |
| INS 8250 Line Control Register | 2-152 |
| INS 8250 Programmable Baud Rate Generator | 2-153 |
| Line Status Register | 2-155 |
| Interrupt Identification Register | 2-157 |
| Interrupt Enable Register | 2-159 |
| Modem Control Register | 2-160 |
| Modem Status Register | 2-161 |
| Receiver Buffer Register | 2-162 |
| Transmitter Holding Register | 2-163 |
| Selecting The Interface Format | 2-164 |
| Asynchronous Communications Adapter Connector Interface Specifications | 2-165 |
| Prototype Card | 2-167 |
| Prototype Card Block Diagram | 2-168 |
| I/O Channel Interface | 2-169 |
| Prototype Card to System Board Interface | 2-171 |
| System Loading and Power Limitations | 2-172 |
| Prototype Card External Interface | 2-173 |

SECTION 3. ROM AND SYSTEM USAGE

| | |
|---|------|
| ROM BIOS | 3-1 |
| Use of BIOS | 3-2 |
| Parameter Passing | 3-2 |
| Interrupt Vector Listing | 3-3 |
| Vectors With Special Meaning | 3-5 |
| Interrupt 1CH - Timer Tick | 3-5 |
| Interrupt 1DH - Video Parameters | 3-5 |
| Interrupt 1EH - Diskette Parameters | 3-5 |
| Interrupt 1FH - Graphics Character Extensions | 3-6 |
| Other Read/Write Memory Usage | 3-6 |
| BIOS Programming Tip | 3-6 |
| BIOS Memory Map | 3-7 |
| BIOS Cassette Logic | 3-8 |
| Interrupt 15 | 3-8 |
| Cassette Write | 3-8 |
| Cassette Read | 3-9 |
| Data Record Architecture | 3-10 |
| Error Recovery | 3-10 |

| | |
|--|-------|
| Keyboard Encoding and Usage | 3-11 |
| Encoding | 3-11 |
| Character Codes | 3-11 |
| Extended Codes | 3-13 |
| Extended Functions | 3-13 |
| Shift States | 3-14 |
| Shift Key Priorities | 3-15 |
| Special Handling | 3-15 |
| System Reset | 3-15 |
| Break | 3-16 |
| Pause | 3-16 |
| Print Screen | 3-16 |
| Keyboard Usage | 3-17 |
| BASIC Screen Editor Special Functions | 3-19 |
| DOS Special Functions | 3-19 |
| Low Memory Maps | 3-21 |
| 0-7F Interrupt Vectors | 3-21 |
| BASIC and DOS Reserved Interrupts (80-3FF) | 3-22 |
| Reserved Memory Locations (400-5FF) | 3-22 |
| BASIC Workspace Variables | 3-23 |
| SECTION 4. APPENDICES | A-0 |
| Appendix A: ROM BIOS Listing | A-1 |
| Appendix B: Assembly Instruction Set Reference | B-1 |
| Appendix C: Of Characters Keystrokes and Color | C-1 |
| Appendix D: Logic Diagrams | D-1 |
| Appendix E: Unit Specifications | E-1 |
| Glossary | G-1 |
| Bibliography | Bib-1 |
| Index | I-1 |

FIGURE LISTING

| | |
|---|-------|
| 1. System Block Diagram | 1-6 |
| 2. System Board Data Flow | 2-6 |
| 3. I/O Channel Diagram | 2-9 |
| 4. System Board Component Diagram | 2-13 |
| 5. Keyboard Interface Block Diagram | 2-15 |
| 6. Keyboard Diagram | 2-16 |
| 7. Cassette Interface Read Hardware | 2-25 |
| 8. Cassette Interface Write Hardware | 2-26 |
| 9. Cassette Motor Control | 2-26 |
| 10. Speaker Drive System Block Diagram | 2-28 |
| 11. System Memory Map | 2-31 |
| 12. System Memory Map (Increments of 16KB) | 2-32 |
| 13. Power Supply and Connectors | 2-45 |
| 14. IBM Monochrome Display Adapter Block Diagram | 2-48 |
| 15. Color/Graphics Monitor Adapter Block Diagram | 2-57 |
| 16. Parallel Printer Adapter Block Diagram | 2-76 |
| 17. Location of (Printer) DIP Switches | 2-83 |
| 18. Parallel Interface Timing | 2-88 |
| 19. 5 1/4" Diskette Drive Adapter Block Diagram | 2-108 |
| 20. Game Control Adapter Block Diagram | 2-135 |
| 21. Joystick Schematic | 2-139 |
| 22. Asynchronous Communications Adapter Block Diagram | 2-142 |
| 23. Current Loop Interface | 2-145 |
| 24. Selecting The Interface Format | 2-164 |
| 25. Prototype Card Block Diagram | 2-168 |
| 26. I/O Channel Interface | 2-169 |
| 27. I/O Channel Interface | 2-170 |
| 28. Prototype Card External Interface | 2-173 |
| 29. BIOS Memory Map | 3-7 |

SECTION 1. HARDWARE OVERVIEW

TABLE LISTING

| | |
|---|-------|
| 1. Keyboard Scan Codes | 2-23 |
| 2. 6845 Initialization Parameters | 2-51 |
| 3. Monochrome Vs Color/Graphics Attributes | 2-61 |
| 4. Color/Graphics Modes | 2-61 |
| 5. Summary of Available Colors | 2-65 |
| 6. 6845 Register Description | 2-66 |
| 7. Printer Specifications | 2-81 |
| 8. Functions and Conditions of DIP Switch 1 | 2-83 |
| 9. Functions and Conditions of DIP Switch 2 | 2-84 |
| 10. Connector Pin Assignment and Description of Interface Signals | 2-86 |
| 11. Graphics Printer Character Set 1 | 2-102 |
| 12. Graphics Printer Character Set 2 | 2-104 |
| 13. Symbol Description | 2-112 |
| 14. Status Register 0 | 2-118 |
| 15. Status Register 1 | 2-119 |
| 16. Status Register 2 | 2-120 |
| 17. Status Register 3 | 2-121 |
| 18. Mechanical and Electrical Specifications | 2-130 |
| 19. Memory Module Pin Configuration | 2-133 |
| 20. DIP Module Start Address | 2-133 |
| 21. I/O Decodes (3F8 - 3FF) | 2-143 |
| 22. Asynchronous Communications Reset Functions | 2-151 |
| 23. BAUD Rate at 1.843 Mhz | 2-155 |
| 24. Interrupt Control Functions (Asynchronous) | 2-158 |
| 25. Character Codes | 3-11 |
| 26. Keyboard Extended Functions | 3-14 |
| 27. Keyboard - Commonly Used Functions | 3-17 |
| 28. Basic Screen Editor Special Functions | 3-19 |
| 29. DOS Special Functions | 3-19 |
| 30. 0-7F Interrupt Vectors | 3-21 |
| 31. Basic & DOS Reserved Interrupts (80-3FF) | 3-22 |
| 32. Reserved Memory Locations (400-5FF) | 3-22 |

The IBM Personal Computer has two major elements; a System Unit and a keyboard. In addition, a variety of options are offered including one or two 5-1/4" Diskette Drives with adapter which can be housed inside the System Unit, an IBM Monochrome Display, an IBM 80 CPS Graphics Printer, two display adapters, storage increments to 544KB, an Asynchronous Communications Adapter, Printer Adapter and a Game Control Adapter.

The System Unit is the heart of your IBM Personal Computer system. The System Unit houses the microprocessor, Read-Only Memory (ROM), Read/Write Memory, Power Supply, and System Expansion Slots for the attachment of up to five options. One or two 5-1/4" Diskette Drives can also be mounted in the System Unit providing 160KB or 320 KB of storage each, depending on the type of drive you have.

The System Board is a large board which fits horizontally in the base of the System Unit and includes the microprocessor, 40KB ROM and 16KB memory. The memory can be expanded in 16KB increments to 64KB. The System Board also includes an enhanced version of the Microsoft BASIC-80 Interpreter without diskette functions. The BASIC Interpreter is included in the ROM. The System Board also permits the attachment of an audio cassette recorder for loading or saving programs and data.

The System Unit power supply is a 63.5 watt, 4 level DC and either 120 Vac or 220/240 Vac unit. It is a switching regulator design, allowing for light weight and high efficiency. Its DC power capacity is rated for an expanded system.

The 5-1/4" Diskette Drive permits the IBM Personal Computer to read, write and store data on 5-1/4" diskettes. Each diskette stores approximately 160KB or 320KB of formatted data. Two of these drives may be installed internally in the System Unit.

The keyboard is attached to the System Unit with a light-weight, coiled cable. The keyboard features 83 keys, and offers commonly used data and word processing functions in a design combining the familiar typewriter and calculator pad layouts.

A base system requires one of two different display adapters, either a Color/Graphics medium resolution Monitor adapter or a high resolution monochrome alphanumeric adapter with a parallel printer adapter.

The Color/Graphics Adapter operates at U.S. NTSC (National Television System Committee) standard frequencies (15,750 Hz horizontal and 60 Hz vertical scan), allowing direct attachment to a variety of home TVs (with a user supplied RF modulator) and monitors.

The Color/Graphic Monitor adapter supports a variety of modes selected by program control. The adapter supports color or black and white alphanumeric modes with line width of 40 or 80 characters and 25 lines. In the alphanumeric mode there are 256 characters.

This adapter provides both a standard composite video and direct drive outputs. In addition, a light pen feature input port is provided.

The IBM Monochrome Display is a high resolution green phosphor display offering the personal computer user quality usually found on larger computer systems. The display features an 11-1/2" screen with an anti-glare surface and a variety of highlighting choices. The screen displays 25 lines of 80 characters. It supports 256 different letters, numbers and special characters that are formed in a nine by 14 dot matrix.

The IBM Monochrome Display requires the Monochrome Display and Printer Adapter Option. This option installs in one of the System Unit's five System Expansion Slots. The display is powered from the System Unit.

The IBM 80 CPS Graphics Printer is a versatile, low cost, quality printer. It prints in both directions at a nominal horizontal speed of 80 characters per second on continuous-feed, single or multipart paper. The printer features four character sizes (40, 66, 80, or 132 characters per line), subscript, superscript, and underlining. It has uppercase and lowercase ASCII and International character sets, a defined graphic character set and programmable graphics. In addition it has a Power-on Self-test, simple paper loading and a cartridge ribbon.

The IBM 80 CPS Graphics Printer requires either the Monochrome Display and Printer Adapter or the Printer Adapter installed in one of the System Unit's five System Expansion Slots. The printer is available in either a 120 volt, 60 Hz version or a 220 or 240 volt, 50/60 Hz version and plugs into a standard wall outlet. The printer requires the Printer Cable Option for attachment to the System Unit.

The 16KB Memory Module Kits allow you to increase the memory size of your IBM Personal Computer. The base system comes standard with 16KB of memory. Up to three 16KB Memory Module Kits may be installed to increase the memory size to 64KB. Memory can be further increased to 544KB with additional memory options once these three Expansion Kits are installed.

The 16KB Memory Kits plug into the System Board and must be installed sequentially. They do not occupy any of the five System Expansion Slots.

The 32KB, 64KB, and 64/256KB Memory Expansion Options permit memory capacity to be increased beyond 64KB. Any combination of up to three Memory Expansion Options may be installed as long as System Expansion Slots are available. A maximum of 480KB of Memory Expansion Options can be addressed, which when added to 64KB on the System Board, gives a total memory capacity of 544KB.

Each 32KB, 64KB, or 64/256KB Memory Expansion Option requires a System Expansion Slot in the System Unit. The first 64KB on the System Board is required before any Memory Expansion Options can be installed.

The 64KB Memory Module Kits allow the memory size of a 64/256KB Memory Expansion Option to be increased. Each 64KB Kit consists of 9 modules which plug into the 64/256KB Option and must be installed sequentially. The base 64/256KB Option comes standard with 64KB of memory. One, two, or three 64KB Kits may be added, providing the 64/256KB Option with 128KB, 192KB, or 256KB of memory respectively.

The Asynchronous Communications Adapter provides a channel to data processing or input/output devices outside of your immediate system. These can be connected by telephone using a plug-in modem, or directly by cable when the device is nearby.

This option utilizes the RS232C asynchronous (start-stop) interface permitting attachment to a variety of devices including a large "host" computer or another IBM Personal Computer.

This option supports 50 to 9600 BPS transmission speeds. One 25 pin "D" shell, male type connector is provided to attach various peripheral devices. A "current loop" interface is located in the same connector, and a jumper block is provided to manually select either the voltage or the current loop interface.

The Asynchronous Communications Adapter requires a System Expansion Slot in the System Unit. Telephone line transmission requires an external modem which is subject to local telephone administration and national regulations. Note that references to local telephone administration and national regulations should not be taken to imply that permission to connect has been, or will be, obtained in any particular country.

The Game Control Adapter permits the attachment of user-supplied Joysticks or paddles. Two Joysticks and up to four paddles may be attached. IBM does not manufacture either the Joysticks or the paddles. This option provides connectors for Joysticks or paddles and requires a System Expansion Slot in the System Unit.

A block diagram of the system is on the following pages (1-6 and 1-7).

The Prototype Card is provided by IBM for those who wish to design a customized option for the IBM Personal Computer. The Prototype Card is inserted into an expansion slot in the System Unit.

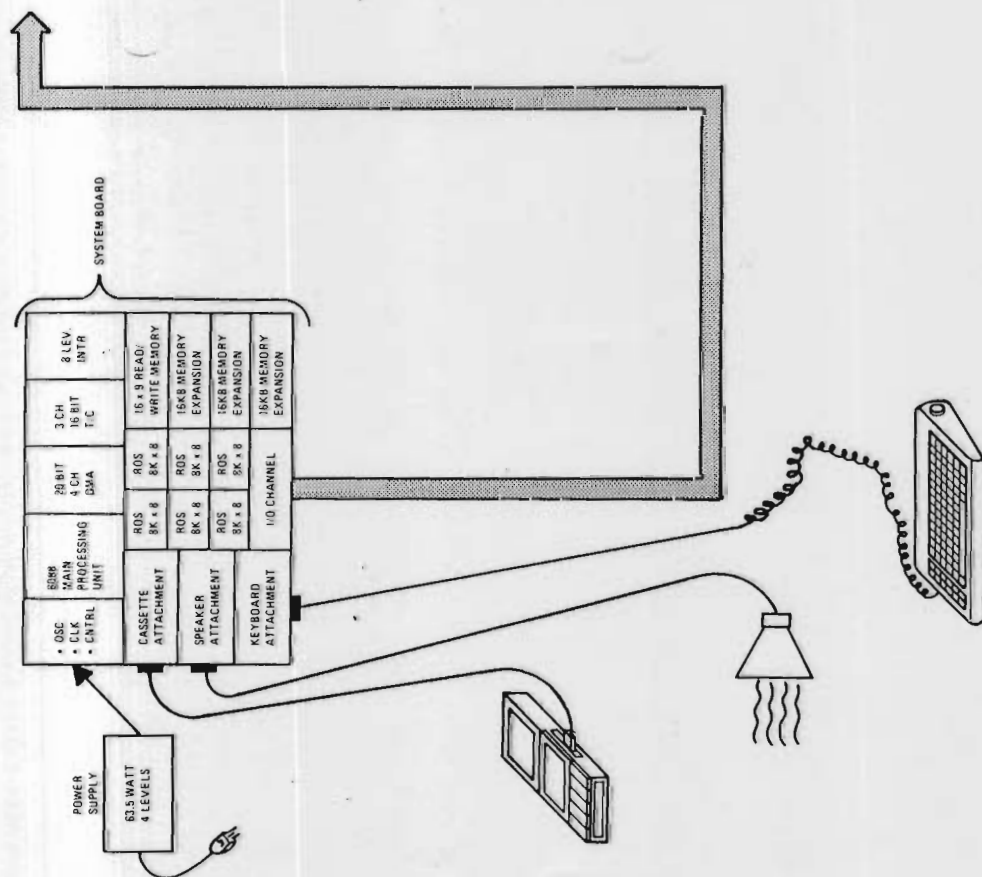


Figure 1. SYSTEM BLOCK DIAGRAM

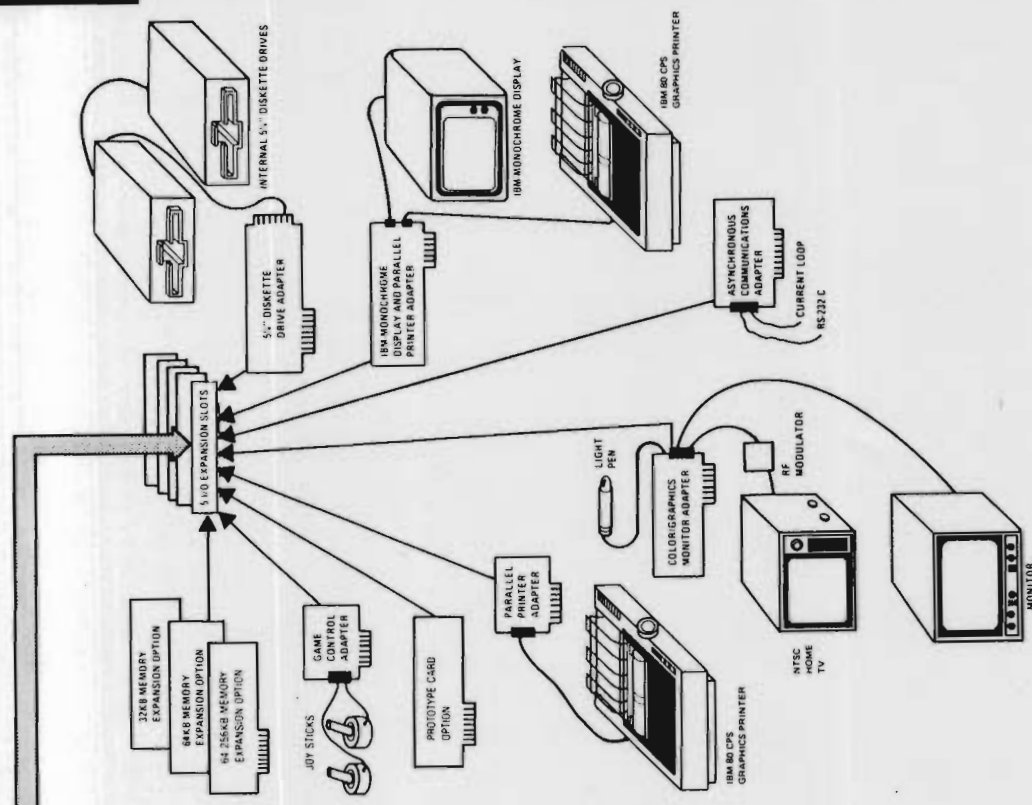


Figure 1. SYSTEM BLOCK DIAGRAM (continued)

System Board

The System Board fits horizontally in the base of the System Unit and has dimensions of approximately 8-1/2 inches by 11 inches. The System Board is a multilayer single land-per-channel design, with ground and power internal planes provided. DC power and a signal from the power supply enter the board through two six pin connectors. Other connectors on the board are for attaching the keyboard, audio cassette, and the speaker. Five 62-pin card edge sockets are also mounted on the System Board. The system I/O channel is bussed across these five I/O slots.

There are 16 (13 used) Dual In-line Package (DIP) switches mounted on the card which can be read under program control. These switches are used to indicate to the system software what options are installed. They are used to indicate amounts of installed storage, both on the System Board and in the System Expansion slots, type of display adapter installed, and desired operation modes upon power-up; ie, color or black and white and 80- or 40 character lines. Switches are also used to identify when the operating system is to be loaded from diskette, and how many diskette drives are attached.

The major elements of the System Board are divided into five major functional areas. They are, the processor subsystem and its support elements, the Read-Only Memory (ROM) subsystem, the Read/Write (R/W) Memory subsystem, integrated I/O adapters, and the I/O channel. All functions are described in detail in this section, except for the I/O channel, which has its own section. Figure 2.0 "System Board Data Flow" page 2-6, illustrates these functional areas.

The heart of the System Board is the Intel 8088 microprocessor. This processor is an 8-bit bus version of the 16-bit 8086 processor by Intel. It is software compatible with the 8086 and, thus, supports 16-bit operations including multiply and divide. The processor supports 20 bits of addressing (1 megabyte of storage). The processor is implemented in maximum mode so a co-processor can be added as a feature. The processor is operated at 4.77 Mhz. This frequency is derived from a 14.31818 Mhz crystal which is divided by three for the processor clock and by four to obtain the 3.58 Mhz color burst signal required for color televisions. At the 4.77 Mhz clock rate, the 8088 bus cycles are four clocks of 210 ns or 840 ns. I/O cycles take five 210 ns clocks or 1.05 microsec (m sec).

SECTION 2. HARDWARE

Contents

| | |
|---|-------|
| System Board | 2-3 |
| Power Supply | 2-43 |
| IBM Monochrome Display And Printer Adapter | 2-47 |
| IBM Monochrome Display | 2-53 |
| Color/Graphics Monitor Adapter | 2-55 |
| Printer Adapter and Printer | 2-75 |
| 5-1/4" Diskette Drive Adapter | 2-107 |
| 5-1/4" Diskette Drive | 2-128 |
| Memory Expansion Options: 32, 64 and 64/256KB | 2-131 |
| Game Control Adapter | 2-135 |
| Asynchronous Communications Adapter | 2-141 |
| Prototype Card | 2-167 |

HARDWARE

HARDWARE

The processor is supported by a set of high function support devices providing four channels of 20-bit Direct Memory Access (DMA), three 16-bit timer counter channels, and eight prioritized interrupt levels.

Three of the four DMA channels are available on the I/O bus and are provided to support high speed data transfers between I/O devices and memory without processor intervention. The fourth DMA channel is programmed to refresh the system dynamic memory. This is done by programming a channel of the timer counter device to periodically request a dummy DMA transfer. This creates a memory read cycle which is available to refresh dynamic storage both on the System Board and in the System Expansion slots. All DMA data transfers, except the refresh channel, take five processor clocks of 210 ns or 1.05 ns if the processor ready line is not deactivated. Refresh DMA cycles take four clocks or 840 ns.

The three timer/counters are used by the system as follows: Channel 0 is used to time and request refresh cycles from the DMA channel, Channel 2 is used to support the tone generation for the audio speaker, and Channel 1 is used by the system as a general purpose timer providing a constant time base for implementing a time-of-day clock. Each channel has a minimum timing resolution of 1.05 μ sec.

Of the eight prioritized levels of interrupt, six are bussed to the I/O slots for use by feature cards. Two levels are used on the System Board. Level 0, the highest priority, is attached to Channel 1 of the timer counter and provides a periodic interrupt. Level 1 is attached to the keyboard adapter circuits and receives an interrupt for each scan code sent by the keyboard. The Non-Maskable Interrupt (NMI) of the 8088 is used to report memory parity errors.

The System Board is designed to support both ROM and Read/Write Memory. The System Board contains space for 48K x 8 of ROM or EPROM. Six module sockets are provided, each capable of accepting an 8K x 8 device. Five of the sockets are populated with 40KB of ROM. This ROM contains the Cassette BASIC interpreter, cassette operating system, Power-on Self-test, I/O drivers, dot patterns for 128 characters in graphics mode, and a diskette bootstrap loader. The ROM is packaged in 24-pin modules and has an access time of 250ns and a cycle time of 375 ns.

The System Board also contains from 16K x 9 to 64K x 9 of Read/Write Memory. A minimum system would have 16KB of memory with module sockets for an additional 48KB. In a cassette version of the system, approximately 4KB is used by the system leaving approximately 12KB of user's space for BASIC programs. Additional memory beyond the System Board's maximum of 64KB, is obtained by adding memory cards in the System Expansion slots.

The memory is dynamic 16K x 1 chips with an access time of 250 ns and a cycle time of 410 ns. All R/W memory is parity checked. The System Board contains circuits for attaching an audio cassette, the serial keyboard, and the speaker. The cassette adapter allows the attachment of any good quality audio cassette via either the microphone or auxiliary inputs. The board has a jumper for either input. This interface also provides a cassette motor control line for transport starting and stopping under program control. This interface reads and writes the audio cassette at a data rate of between 1,000 and 2,000 baud. The baud rate is variable and dependent on data content since a different bit-cell time is used for 0's and 1's. For diagnostic purposes, the tape interface can loop read to write to test the board's circuits. The system software blocks cassette data, generates and checks data with a Cyclic Redundancy Check (CRC).

The processor also contains the adapter circuits for attaching the serial interface from the keyboard. This generates an interrupt to the processor when a complete scan code is received. This interface can request execution of a diagnostic in the keyboard.

Both the keyboard and cassette interfaces are provided via 5-pin DIN connectors, which are right angle mounts on the System Board and extend through the rear panel of the System Unit.

The system is provided with a 2-1/4 inch audio speaker mounted inside the System Unit. The System Board contains the control circuits and driver for the speaker. The speaker connects through a 2-wire interface which attaches to a 4-pin header on the System Board.

The speaker drive circuit is capable of approximately a 1/2 watt of power. The control circuits allow the speaker to be driven several different ways. First, a direct program control register bit may be toggled to generate a pulse train; second, the output of Channel 2 of the timer counter may be programmed to generate a waveform to the speaker. Third, the clock input to the timer/counter can be modulated with a program controlled I/O Register Bit. All three forms of control may be performed simultaneously.

System Board Data Flow

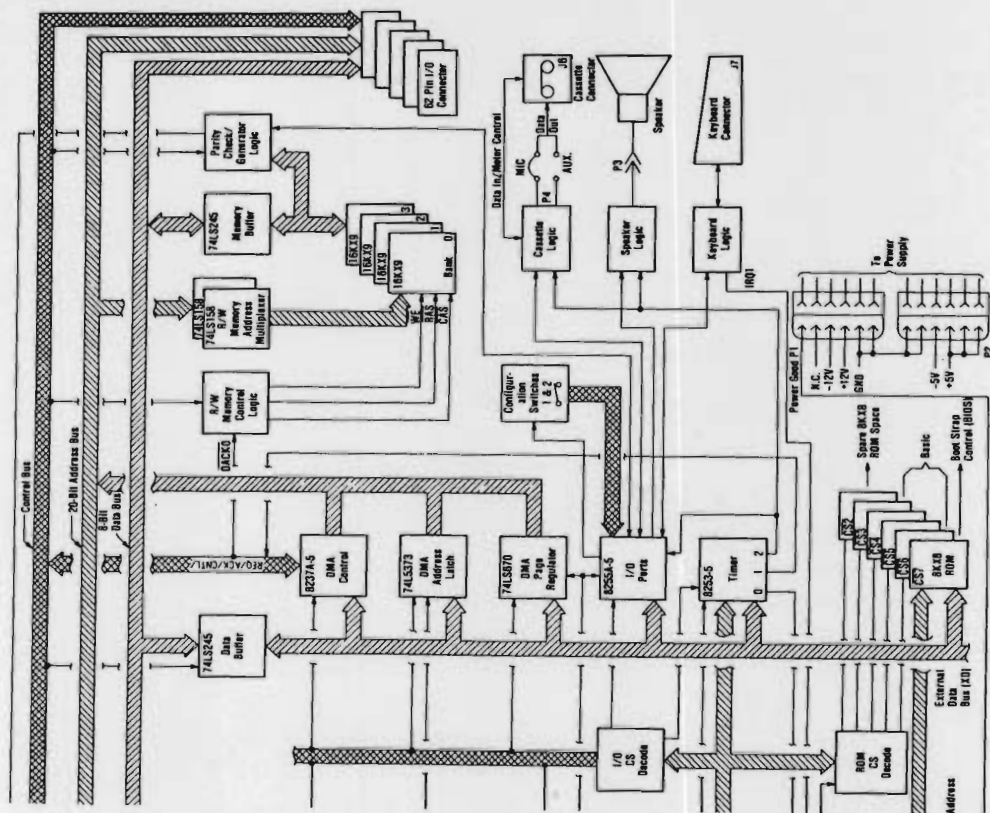


Figure 2. SYSTEM BOARD DATA FLOW (SHEET 2)

I/O Channel

The I/O channel is an extension of the 8088 microprocessor bus. It is, however, demultiplexed, repowered, and enhanced by the addition of interrupts and Direct Memory Access (DMA) functions.

The I/O channel contains an 8-bit bidirectional data bus, 20 address lines, 6 levels of interrupt, control lines for memory and I/O read or write, clock and timing lines, 3 channels of DMA control lines, memory refresh timing control lines a channel check line, and power and ground for the adapters. Four voltage levels are provided for I/O card +5 Vdc, -5 Vdc, +12 Vdc, and -12 Vdc. These functions are provided in a 62-pin connector with 100 mil card tab spacing.

A ready line is available on the I/O channel to allow operation with slow I/O or memory devices. If the channel's Ready line is not activated by an addressed device, all processor generated memory read and write cycles take four 210 ns clock or 840 ns/byte. All processor-generated I/O read and write cycles require five 210 ns clocks or 1.05 m sec/byte. All DMA transfers require five clocks for a cycle time of 1.05 m sec/byte. Refresh cycles are present once every 72 clocks or approximately 15 m sec and require five clocks or approximately 7% of the bus bandwidth. I/O devices are addressed using I/O mapped address space. The channel is designed so that 512 I/O device addresses are available to the I/O channel cards.

A channel check line exists for reporting error conditions to the processor. Activating this line results in a NMI to the 8088 processor. Memory Expansion Options use this line to report parity errors.

The I/O channel is repowered so there is sufficient drive to power all five System Expansion Slots, assuming two loads per slot. The IBM Option I/O adapters typically use only one load. A graphic illustration of the System I/O Channel and its descriptions are on the following pages.

I/O Channel Diagram

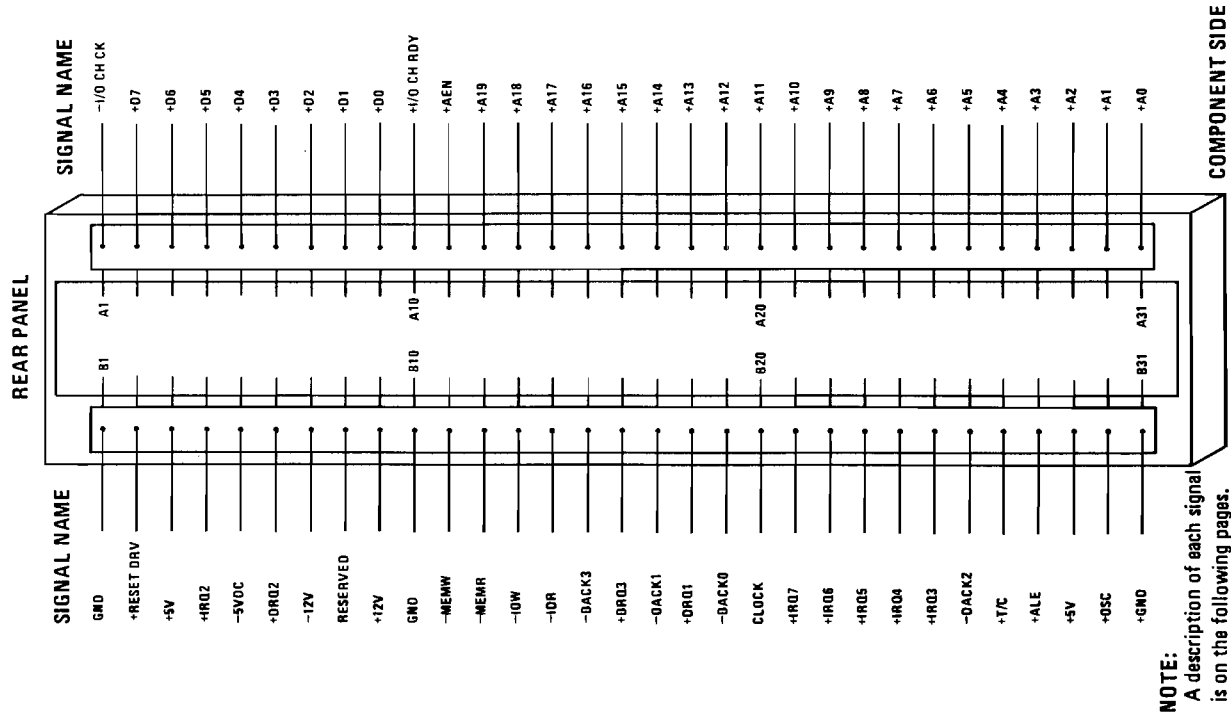


Figure 3. I/O CHANNEL DIAGRAM

System Board I/O Channel Description

The following is a description of the IBM Personal Computer System Board I/O Channel. All signal lines are TTL compatible.

| Signal | I/O | Description |
|------------------|-----|--|
| OSC | O | Oscillator: This signal is a high speed clock with a 70 nsec. period (14.31818 MHz). It has a 50% duty cycle. |
| CLK | O | Clock: This is the system clock. It is a divide - by - three of the oscillator and has a period of 210 nsec. (4.77 Mhz.) The clock has a 33% duty cycle. |
| RESET DRV | O | Reset Driver: This line is used to reset or initialize system logic upon power-up or during a low line voltage outage. This signal is synchronized to the falling edge of clock and is active HIGH. |
| A0-A19 | O | Address Bits 0 to 19: These lines are used to address memory and I/O devices within the system. The 20 address lines allow access of up to 1 megabyte of memory. A0 is the Least Significant Bit (LSB) while A19 is the Most Significant Bit (MSB). These lines are generated by either the processor or the DMA Controller. They are active HIGH. |
| D0-D7 | I/O | Data Bits 0 to 7: These lines provide data bus bits 0 to 7 for the processor, memory, and I/O Devices. D0 is the Least Significant Bit (LSB) and D7 is the Most Significant Bit (MSB). These lines are active HIGH. |
| ALE | O | Address Latch Enable: This is provided by the 8288 Bus Controller and is used on the System Board to latch valid addresses from the processor. It is available to the I/O Channel as an indicator of a valid processor address (when used in conjunction with AEN). Processor addresses are latched with the falling edge of ALE. |
| <u>I/O CH CK</u> | I | -I/O Channel Check: This line provides the CPU with parity (error) information on memory or devices in the I/O Channel. When this signal is active LOW, a parity error is indicated. |

| Signal | I/O | Description |
|-------------------|-----|--|
| <u>I/O CH RDY</u> | I | I/O Channel Ready: This line (normally high or "READY") is pulled low ("NOT READY") by a memory or I/O device to lengthen I/O or memory cycles. It allows slower devices to attach to the I/O Channel with a minimum of difficulty. Any slow device using this line should drive it low immediately upon detecting a valid address and a Read or Write command. This line should never be held low for any period in excess of 10 clock cycles (2.1 usec.) Machine cycles (I/O or memory) are extended by an integral number of CLK cycles (210 ns). |
| <u>IRQ2-IRQ7</u> | I | Interrupt Request 2 to 7: These lines are used to signal the processor that an I/O device requires attention. They are prioritized with IRQ2 as the highest priority and IRQ7 as the lowest. An Interrupt Request is generated by raising an IRQ line (Low to High) and holding it high until it is acknowledged by the processor (Interrupt Service Routine). |
| <u>IOR</u> | O | -I/O Read Command: This command line instructs an I/O device to drive its data onto the data bus. It may be driven by the processor or the DMA Controller. This signal is active LOW. |
| <u>IOW</u> | O | -I/O Write Command: This command line instructs an I/O device to read the data on the data bus. It may be driven by the processor or the DMA Controller. This signal is active LOW. |
| <u>MEMR</u> | | -Memory Read Command: This command line instructs the memory to drive its data onto the data bus. It may be driven by the processor or the DMA Controller. This signal is active LOW. |
| <u>MEMW</u> | O | -Memory Write Command: This command line instructs the memory to store the data present on the data bus. It may be driven by the processor or the DMA Controller. This signal is active LOW. |

Signal I/O Description

DRQ1-DRQ3 I DMA Request 1 to 3: These lines are asynchronous channel requests used by peripheral devices to gain DMA service. They are prioritized with DRQ1 having highest priority and DRQ3 the lowest. A request is generated by bringing a DRQ line to an active level (HIGH). A DRQ line must be held high until the corresponding DACK line goes active.

DACK0-DACK3 O DMA Acknowledge 0 to 3: These lines are used to acknowledge DMA requests (DRQ1-DRQ3) and to refresh system dynamic memory (DACK0). They are active LOW.

AEN O Address Enable: This line is used to degate the processor and other devices from the I/O Channel to allow Direct Memory Access (DMA) transfers to take place. When this line is active (HIGH), the DMA Controller has control of the address bus, data bus, read command lines, (memory and I/O), and the write command lines, (memory and I/O).

T/C O Terminal Count: This line provides a pulse when the terminal count for any DMA channel is reached. This signal is active HIGH.

The following voltages are available on the System Board I/O Channel:

- +5 Vdc \pm 5%, Located on 2 connector pins.
- 5 Vdc \pm 10%, Located on 1 connector pin.
- +12 Vdc \pm 5%, Located on 1 connector pin.
- 12 Vdc \pm 10%, Located on 1 connector pin.
- GND (Ground), Located on 3 connector pins.

System Board Component Diagram

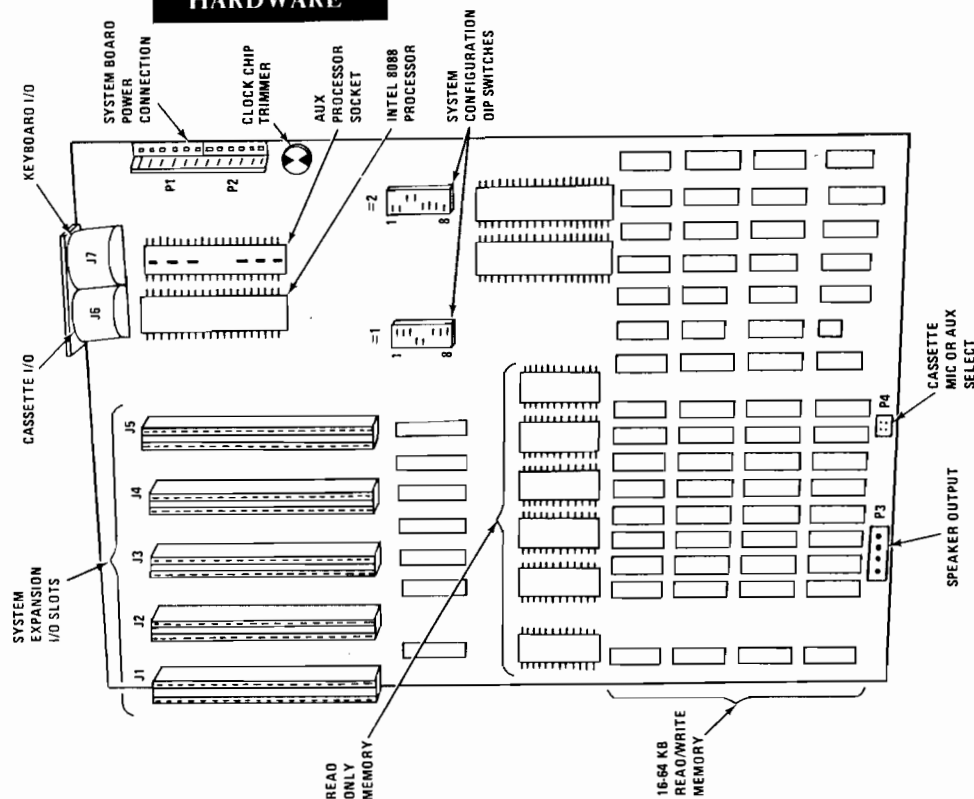


Figure 4. SYSTEM BOARD COMPONENT DIAGRAM

Keyboard

The Keyboard is a device separate from the System Unit. It is attached via a serial interface cable approximately 6 feet in length which plugs into the rear of the System Unit. The attaching cable is coiled, like that of a telephone headset, and is a shielded four-wire wire connection. The interface contains power (+5 Vdc), ground and two bidirectional signal lines. The cable is permanently attached at the keyboard end and plugs into the System Unit via a DIN connector.

The keyboard uses a capacitive technology with a microcomputer (Intel 8048) performing the keyboard scan function. The keyboard interface is defined so system software has the maximum flexibility in defining keyboard operations such as shift states of keys, make/break keys, and typematic operation. This is accomplished by having the keyboard return scan codes rather than American Standard Code for Information Interchange (ASCII) codes. In addition, all keys except control keys are typematic and generate both a make and a break scan code. For example, key 1 produces scan code 01 on make, and code 81 on break. Break codes are formed by adding X'80' to make codes. The keyboard I/O driver can define keyboard keys as shift or typematic keys as required by the application.

The microcomputer (Intel 8048) in the keyboard performs several functions including a Power-on Self-test and when requested by the System Unit. This diagnostic CRC checks the microcomputer ROM, tests memory and checks for stuck keys. Additional functions are: keyboard scanning, key debounce, buffering of up to 20 key scan codes, maintaining bidirectional serial communications with the System Unit, and executing the hand shake protocol required by each scan code transfer. A keyboard diagram and table of scan codes are on the following pages. Figure (5) is a block diagram of the keyboard interface on the System Board.

Several different keyboard arrangements are available. These are illustrated on the following pages together with a table of scan codes for the U.S. keyboard. For information on the keyboard routines required to implement non-U.S. keyboards, refer to the Guide to Operations and DOS manuals.

Note: Reference to local telephone administration and national regulations should not be taken to imply that permission to connect has been, or will be, obtained in any particular country.

Keyboard Interface Block Diagram

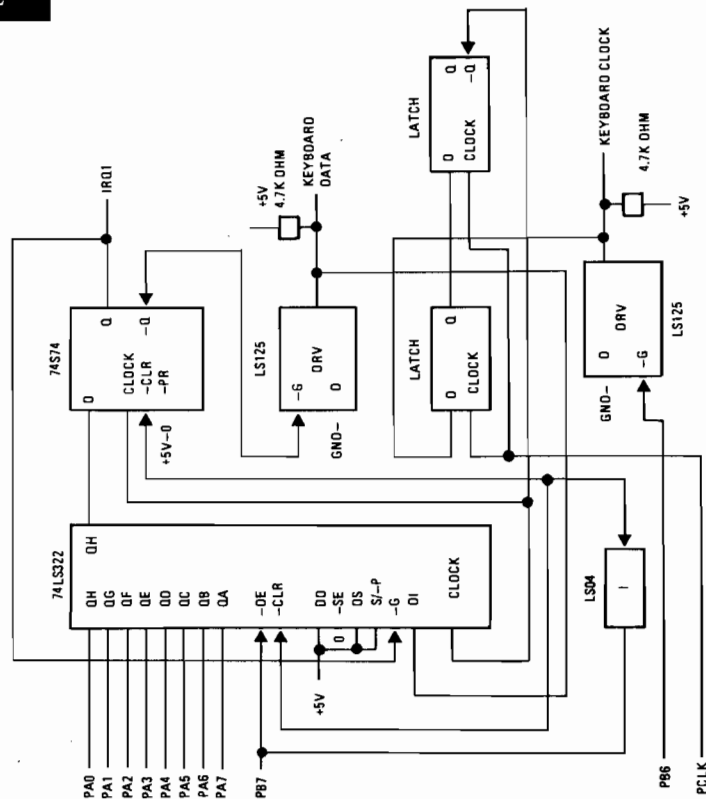
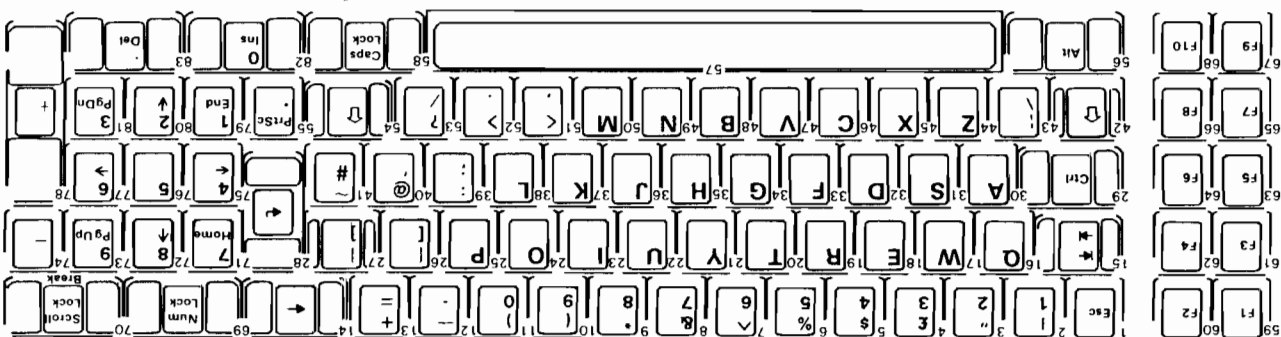


Figure 5. KEYBOARD INTERFACE BLOCK DIAGRAM

Figure 6. KEYBOARD DIAGRAM (Continued)

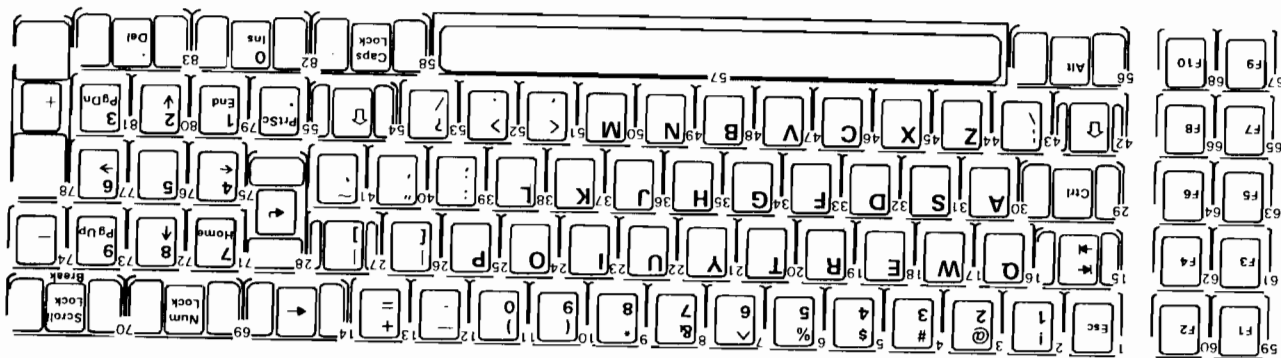
NOTE
1 NOMENCLATURE IS ON BOTH TOP AND FRONT FACE OF KEYBUTTON AS SHOWN.
THE NUMBER TO THE UPPER LEFT DESIGNATES THE BUTTON POSITION.



United Kingdom Keyboard Diagram

Figure 6. KEYBOARD DIAGRAM

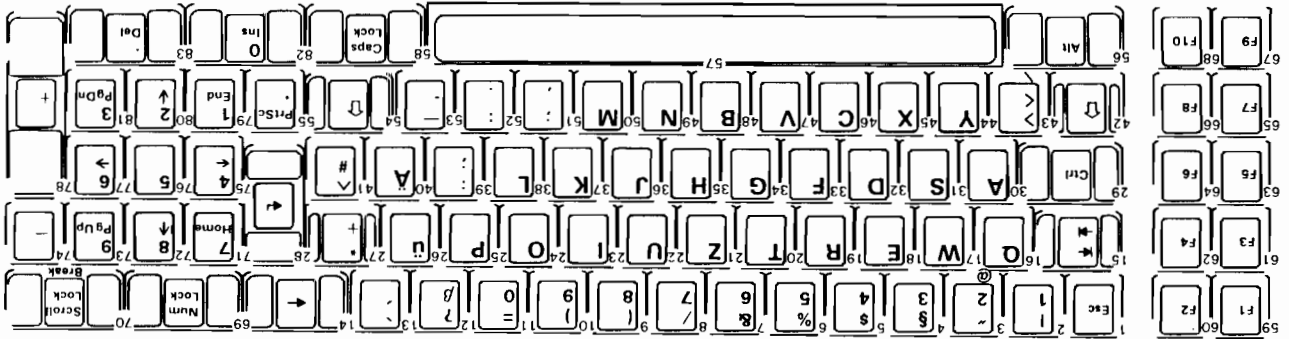
NOTE
1 NOMENCLATURE IS ON BOTH TOP AND FRONT FACE OF KEYBUTTON AS SHOWN.
THE NUMBER TO THE UPPER LEFT DESIGNATES THE BUTTON POSITION.



United States Keyboard Diagram

Figure 6 (Continued). KEYBOARD DIAGRAM

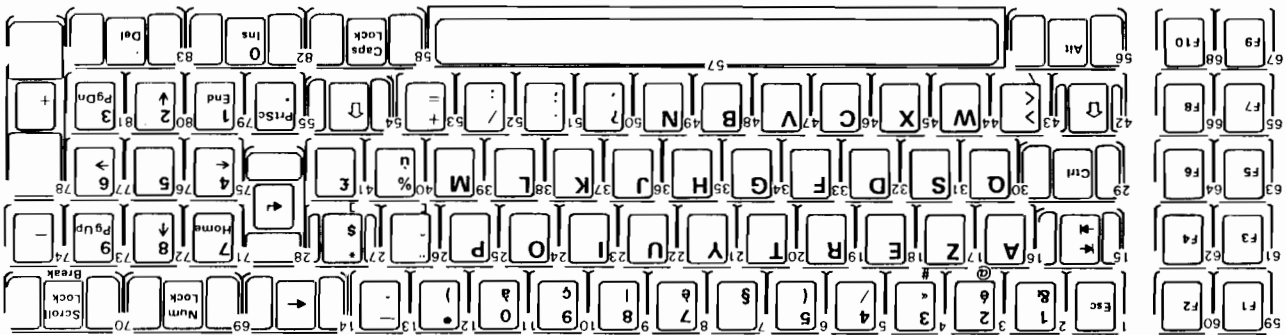
NOTE
1 NOMENCLATURE IS ON BOTH TOP AND FRONT FACE OF KEYBUTTON AS SHOWN.
THE NUMBER TO THE UPPER LEFT DESIGNATES THE BUTTON POSITION.



German Keyboard Diagram

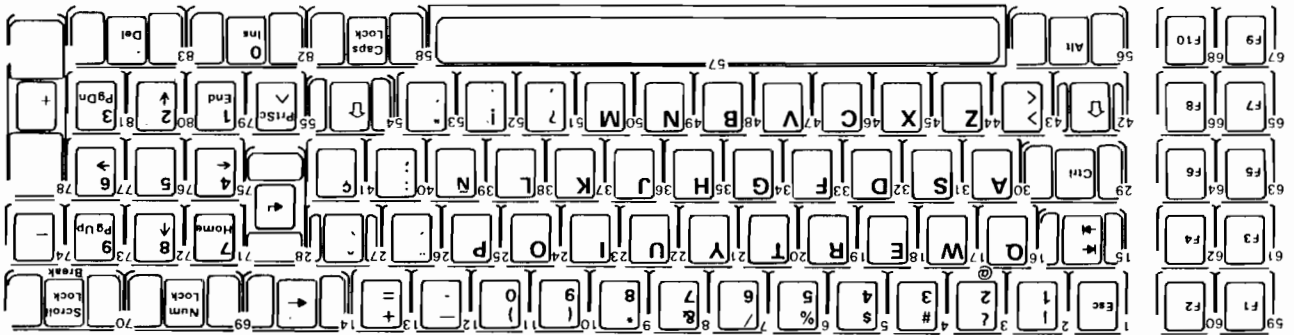
Figure 6. KEYBOARD DIAGRAM (Continued)

NOTE
1 NOMENCLATURE IS ON BOTH TOP AND FRONT FACE OF KEYBUTTON AS SHOWN.
THE NUMBER TO THE UPPER LEFT DESIGNATES THE BUTTON POSITION.



French Keyboard Diagram

NOTE
1 NOMENCLATURE IS ON BOTH TOP AND FRONT FACE OF KEYBUTTON AS SHOWN.
THE NUMBER TO THE UPPER LEFT DESIGNATES THE BUTTON POSITION.



NOTE
1 NOMENCLATURE IS ON BOTH TOP AND FRONT FACE OF KEYBUTTON AS SHOWN.
THE NUMBER TO THE UPPER LEFT DESIGNATES THE BUTTON POSITION.

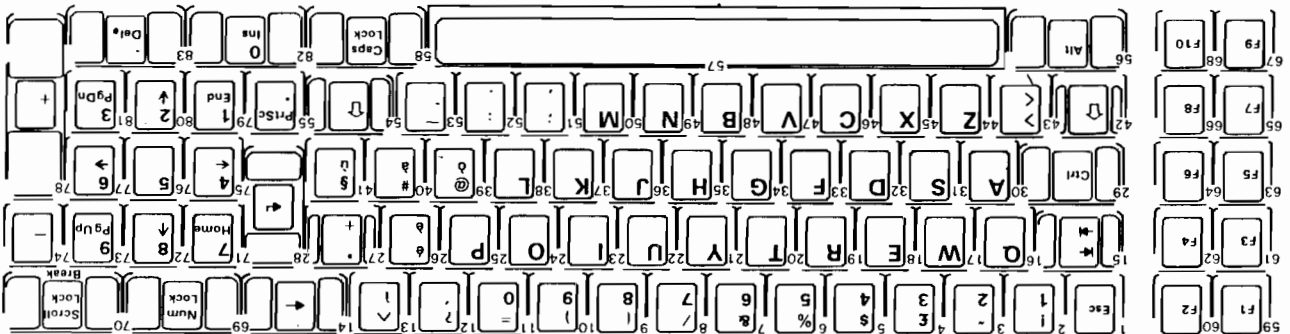
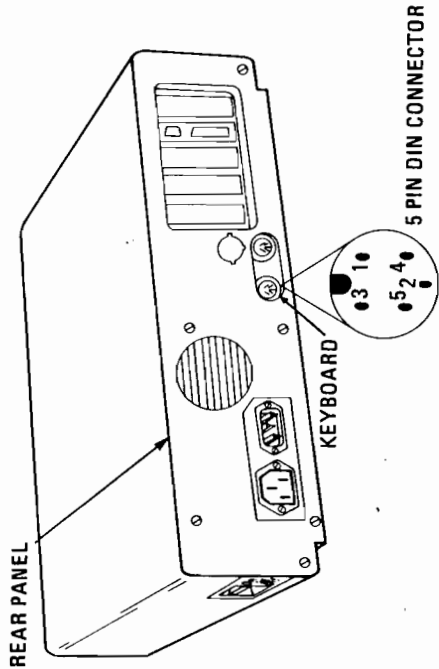


Table 1. Keyboard Scan Codes

| Key Position | Scan Code in Hex | Key Position | Scan Code in Hex |
|--------------|------------------|--------------|------------------|
| 1 | 01 | 43 | 2B |
| 2 | 02 | 44 | 2C |
| 3 | 03 | 45 | 2D |
| 4 | 04 | 46 | 2E |
| 5 | 05 | 47 | 2F |
| 6 | 06 | 48 | 30 |
| 7 | 07 | 49 | 31 |
| 8 | 08 | 50 | 32 |
| 9 | 09 | 51 | 33 |
| 10 | 0A | 52 | 34 |
| 11 | 0B | 53 | 35 |
| 12 | 0C | 54 | 36 |
| 13 | 0D | 55 | 37 |
| 14 | 0E | 56 | 38 |
| 15 | 0F | 57 | 39 |
| 16 | 10 | 58 | 3A |
| 17 | 11 | 59 | 3B |
| 18 | 12 | 60 | 3C |
| 19 | 13 | 61 | 3D |
| 20 | 14 | 62 | 3E |
| 21 | 15 | 63 | 3F |
| 22 | 16 | 64 | 40 |
| 23 | 17 | 65 | 41 |
| 24 | 18 | 66 | 42 |
| 25 | 19 | 67 | 43 |
| 26 | 1A | 68 | 44 |
| 27 | 1B | 69 | 45 |
| 28 | 1C | 70 | 46 |
| 29 | 1D | 71 | 47 |
| 30 | 1E | 72 | 48 |
| 31 | 1F | 73 | 49 |
| 32 | 20 | 74 | 4A |
| 33 | 21 | 75 | 4B |
| 34 | 22 | 76 | 4C |
| 35 | 23 | 77 | 4D |
| 36 | 24 | 78 | 4E |
| 37 | 25 | 79 | 4F |
| 38 | 26 | 80 | 50 |
| 39 | 27 | 81 | 51 |
| 40 | 28 | 82 | 52 |
| 41 | 29 | 83 | 53 |
| 42 | 2A | | |

Keyboard Interface Connector Specifications



| PIN | SIGNAL |
|-----|---|
| 1 | + Keyboard Clock |
| 2 | + Keyboard Data |
| 3 | - Keyboard Reset (Not used by keyboard) |
| 4 | Ground |
| 5 | +5 Volts |

Cassette User Interface

The cassette interface control is implemented in software. (See FIRMWARE Section). An 8253 timer output is used to control the data to the cassette recorder. This output exits the System Board, at the rear, through pin 5 of a DIN connector. The cassette input data is read by an 8255A-5 Programmable Peripheral Interface (PPI) input port bit. This signal is received through pin 4 of the cassette connector. Software algorithms are used to generate and read cassette data. The cassette drive motor is controlled through pins 1 & 3 of the cassette connector. The motor on/off is controlled by an 8255A-PPI output port bit. (Port '61H', bit 3). The 8255A address and bit assignments are defined in the I/O Address Map page. On the following pages are read, write, and motor control block diagrams.

Cassette Jumpers

A 2x2 Berg Pin and Jumper are used on the cassette Data Out line. The jumper will allow the Data Out line to be used as a microphone input (75 mv.) when the jumper is placed across M and C pins. An auxiliary input is available when the jumper is placed across the A and C pins. The auxiliary input provides a .68 volt input to the recorder. Refer to System Board Component Diagram page (2-13) for cassette jumper location.



JUMPER DIAGRAM

Circuit Block Diagrams

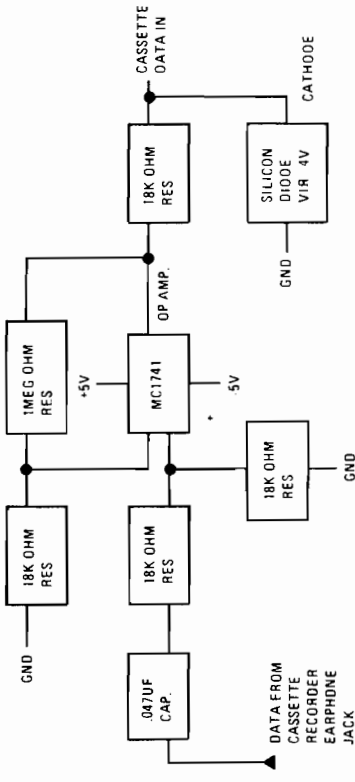
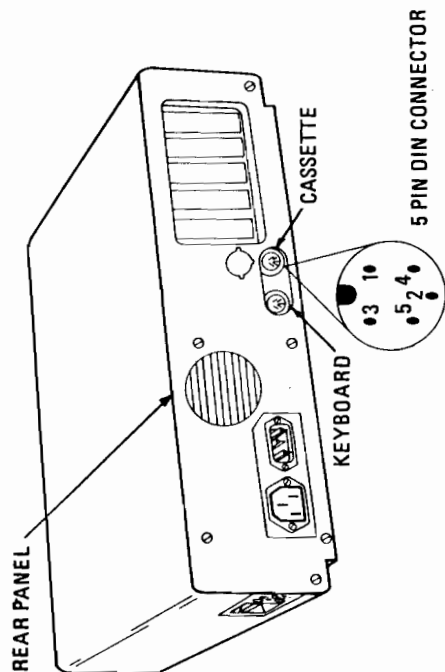


Figure 7. CASSETTE INTERFACE READ HARDWARE

Cassette Interface Connector Specifications



| PIN | SIGNAL | ELECTRICAL CHARACTERISTICS* |
|-----|-----------------------|--|
| 1 | Motor Control | Common from Relay |
| 2 | Ground | |
| 3 | Motor Control | 6 VDC; 1A (Relay N.O.) |
| 4 | Data In | 500nA at $\pm 13V$ - at 1,000 - 2,000 Baud |
| 5 | Data Out (Mic or Aux) | 250 μA at $\left. \begin{array}{l} .68V \\ \text{or} \\ 75mV \end{array} \right\}$ ** |

*All voltages and currents are maximum ratings and should not be exceeded.

**Data out can be chosen using a jumper located on planar.
(AUX \rightarrow .68V or MIC \rightarrow 75 mV).

Interchange of these voltages on the cassette recorder could lead to damage of recorder inputs.

Figure 8. CASSETTE INTERFACE WRITE HARDWARE

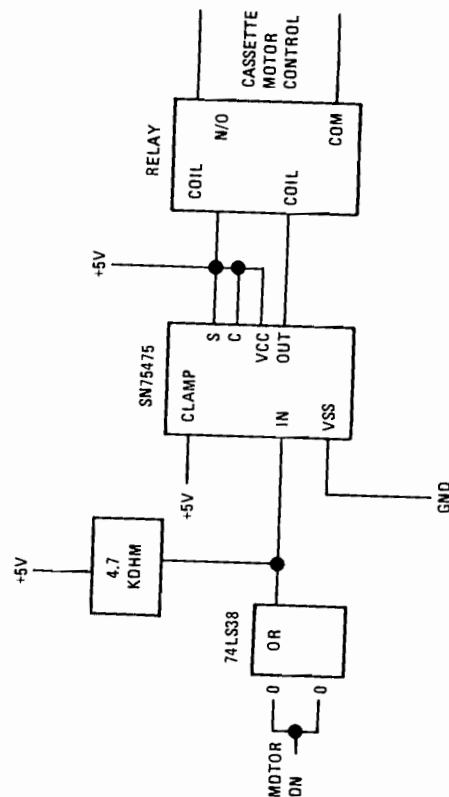
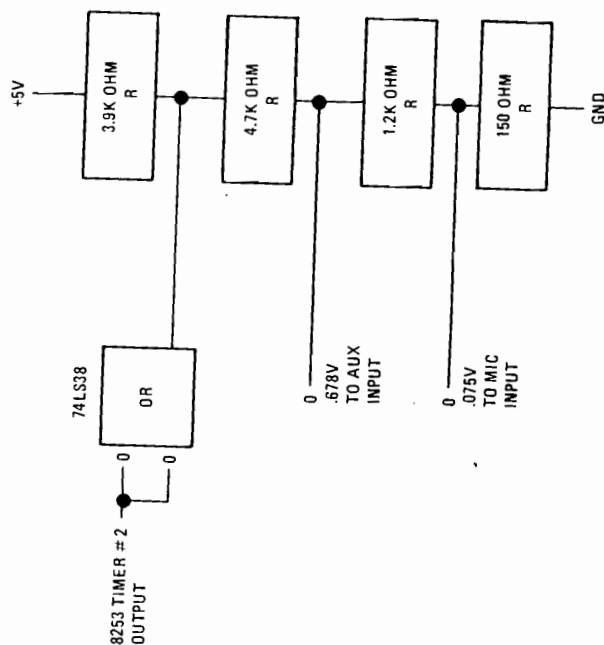


Figure 9. CASSETTE MOTOR CONTROL

Speaker Interface

The sound system contains a small permanent magnet 2-1/4" speaker. The speaker can be driven from one or both of two sources. The sources are:

1. An 8255A-5 PPI output bit. The address and bit are defined in the I/O Address Map pages 2-23 and 2-24.
2. A timer Channel Clock out where the output is programmable within the functions of the 8253-5 timer with a 1.19 Mhz clock input. The timer gate is also controlled by an 8255A-5 PPI output port bit. Address and bit assignment are in the I/O Address Map pages 2-23 and 2-24.

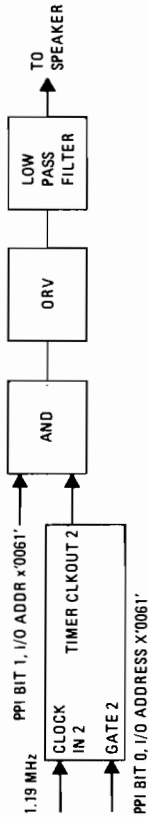


Figure 10. SPEAKER DRIVE SYSTEM BLOCK DIAGRAM

Channel 2 (Tone generation for Speaker)

GATE 2 — Controlled by 8255A-5 PPI Bit
(See I/O Map)

CLK IN 2 — 1.19318 Mhz OSC

CLK OUT 2 — Used to drive Speaker

— Used to write data on the Audio
Cassette

Speaker Connection - 4 Pin Berg Connector, Refer to
System Board Diagram page 2-13 for speaker connection.

| PIN | FUNCTION |
|-----|----------|
| 1 | DATA |
| 2 | KEY |
| 3 | GROUND |
| 4 | +5 VOLTS |

I/O Address Map

| HEX RANGE | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | DEVICE |
|-----------|---|---|---|---|---|----|----|----|----|----|--------------------------|
| 00-0F | 0 | 0 | 0 | 0 | 0 | Z | A3 | A2 | A1 | A0 | DMA CHIP 8237-2 |
| 20-2F | 0 | 0 | 0 | 0 | 0 | Z | Z | Z | Z | A0 | INTERRUPT 8259A |
| 40-4F | 0 | 0 | 0 | 0 | 1 | Z | Z | Z | A1 | A0 | TIMER 8253-5 |
| 60-6F | 0 | 0 | 0 | 0 | 1 | Z | Z | Z | A1 | A0 | PPI 8255A-5 |
| 80-8F | 0 | 0 | 0 | 1 | 0 | Z | Z | Z | A1 | A0 | DMA PAGE REGS |
| *AX | 0 | 0 | 1 | 0 | 0 | Z | Z | Z | A1 | A0 | NMI MASK REG |
| CX | 0 | 0 | 1 | 1 | 0 | Z | Z | Z | A1 | A0 | RESERVED |
| EX | 0 | 0 | 1 | 1 | 1 | Z | Z | Z | A1 | A0 | RESERVED |
| 200-20F | 0 | 0 | 0 | 0 | 0 | 0 | A3 | A2 | A1 | A0 | GAME I/O ADAPTER |
| 278-27F | 1 | 0 | 0 | 1 | 1 | 1 | 1 | Z | A1 | A0 | RESERVED |
| 2F8-2FF | 1 | 0 | 1 | 1 | 1 | 1 | 1 | A2 | A1 | A0 | ALT RS232 CARD |
| 300-31F | 1 | 1 | 0 | 0 | 0 | A4 | A3 | A2 | A1 | A0 | PROTOTYPE CARD |
| 378-37F | 1 | 1 | 0 | 0 | 1 | 1 | 1 | Z | A1 | A0 | PARALLEL PRINTER PORT |
| 280-38F | 1 | 1 | 1 | 0 | 1 | 1 | A3 | A2 | A1 | A0 | IBM MONOCHROME DISPLAY |
| 300-3DF | 1 | 1 | 1 | 1 | 0 | 1 | A3 | A2 | A1 | A0 | PARALLEL PRINTER ADAPTER |
| 3F0-3F7 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | A2 | A1 | A0 | COLOR/GRAPHICS ADAPTER |
| 3F8-3FF | 1 | 1 | 1 | 1 | 1 | 1 | 1 | A2 | A1 | A0 | 5-1/4" DRIVE ADAPTER |
| | | | | | | | 1 | A2 | A1 | A0 | TP RS232 CARD |

Z = Don't Care, i.e., Not in Decode

* At power on time, the Non Mask Interrupt NMI into the 8088 is masked off. This mask bit can be set and reset via system software as follows:

Set mask: write X'80' to I/O Address X'A0' (enable NMI)

Clear mask: write X'00' to I/O Address X'A0' (disable NMI)

I/O Address Map

| I/O Address | +K80 SCAN CODE | | | | | | | | IPL 5 1/4 DISKETTE DRIVE | | | | | | | |
|-------------|----------------|---|---|---|---|---|---|---|--------------------------|---|---|---|---|---|---|---|
| | PA0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| X'0060' | I | N | P | U | T | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |

| I/O Address | +TIMER 2 GATE SPEAKER | | | | | | | | +READ/WRITE MEMORY SIZE OR (READ SPARE KEY) | | | | | | | |
|-------------|-----------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P80 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| X'0061' | O | U | T | P | U | T | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |

| I/O Address | +CASSSETTE DATA IN | | | | | | | | +TIMER CHANNEL 2 OUT | | | | | | | |
|-------------|--------------------|---|---|---|---|---|---|---|----------------------|---|---|---|---|---|---|---|
| | PC0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| X'0062' | I | N | P | U | T | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |

| I/O Address | +READ/WRITE MEMORY PCK | | | | | | | | +READ/WRITE MEMORY PCK | | | | | | | |
|-------------|------------------------|---|---|---|---|---|---|---|------------------------|---|---|---|---|---|---|---|
| | PC0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| X'0063' | I | N | P | U | T | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |

| I/O Address | +READ/WRITE MEMORY PCK | | | | | | | | +READ/WRITE MEMORY PCK | | | | | | | |
|-------------|------------------------|---|---|---|---|---|---|---|------------------------|---|---|---|---|---|---|---|
| | PC0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| X'0064' | I | N | P | U | T | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |

| I/O Address | +READ/WRITE MEMORY PCK | | | | | | | | +READ/WRITE MEMORY PCK | | | | | | | |
|-------------|------------------------|---|---|---|---|---|---|---|------------------------|---|---|---|---|---|---|---|
| | PC0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| X'0065' | I | N | P | U | T | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |

| I/O Address | +READ/WRITE MEMORY PCK | | | | | | | | +READ/WRITE MEMORY PCK | | | | | | | |
|-------------|------------------------|---|---|---|---|---|---|---|------------------------|---|---|---|---|---|---|---|
| | PC0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| X'0066' | I | N | P | U | T | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |

| I/O Address | +READ/WRITE MEMORY PCK | | | | | | | | +READ/WRITE MEMORY PCK | | | | | | | |
|-------------|------------------------|---|---|---|---|---|---|---|------------------------|---|---|---|---|---|---|---|
| | PC0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| X'0067' | I | N | P | U | T | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |

| I/O Address | +READ/WRITE MEMORY PCK | | | | | | | | +READ/WRITE MEMORY PCK | | | | | | | |
|-------------|------------------------|---|---|---|---|---|---|---|------------------------|---|---|---|---|---|---|---|
| | PC0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| X'0068' | I | N | P | U | T | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |

System Memory Map

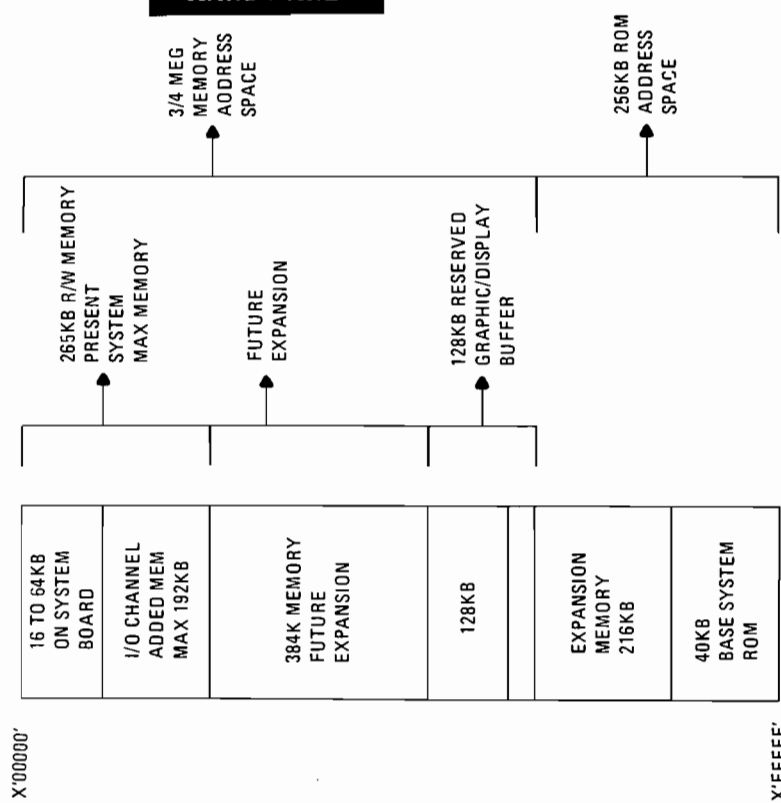


Figure 11. SYSTEM MEMORY MAP

NOTE: PA bit=0 implies switch "ON".
PA bit=1 implies switch "OFF".

System Memory Map (Increments of 16KB)

| START ADDRESS: DECIMAL HEX | FUNCTION: |
|--|---|
| 0 00000 16K 04000 32K 08000 48K 0C000 | 16-64 KB READ/WRITE MEMORY ON SYSTEM BOARD |
| 64K 10000 80K 14000 96K 18000 112K 1C000 | UP TO 192 KB MEMORY IN I/O CHANNEL |
| 128K 20000 144K 24000 160K 28000 178K 2C000 | |
| 192K 30000 208K 34000 224K 38000 240K 3C000 | |
| 256K 40000 272K 44000 288K 48000 304K 4C000 | |
| 320K 50000 336K 54000 352K 58000 368K 5C000 | |
| 384K 60000 400K 64000 416K 68000 432K 6C000 | |
| 448K 70000 464K 74000 480K 78000 496K 7C000 | |
| 512K 80000 528K 84000 544K 88000 560K 8C000 | |
| 576K 90000 592K 94000 608K 98000 624K 9C000 | 384 KB FUTURE R/W MEMORY EXPANSION IN I/O CHANNEL |

Figure 12. SYSTEM MEMORY MAP (INCREMENTS OF 16KB) (SHEET 1 OF 2)

System Memory Map Cont.

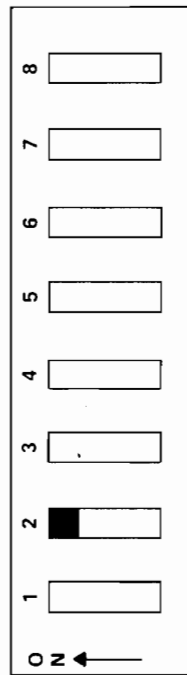
| START ADDRESS: DECIMAL HEX | FUNCTION: |
|--|--|
| 640K A0000 | RESERVED |
| 656K A4000 672K A8000 688K AC000 | 112 KB GRAPHICS/DISPLAY VIDEO BUFFER |
| 704K 80000 | |
| 720K 84000 | |
| 736K 88000 | |
| 752K 8C000 | COLOR/GRAPHICS |
| 768K C0000 784K C4000 800K C8000 816K CC000 | 192 KB MEMORY EXPANSION AREA |
| 832K D0000 848K D4000 864K D8000 880K DC000 | |
| 896K E0000 912K E4000 928K E8000 944K EC000 | |
| 960K F0000 | |
| 976K F4000 992K F8000 1,008M FC000 | |
| | RESERVED |
| | 48 KB BASE SYSTEM ROM |

Figure 12. SYSTEM MEMORY MAP (16KB) (SHEET 2)

System Board and Memory Expansion Switch Settings

On the following four pages are graphic illustrations of switch settings. These are necessary for the system to address components attached, and to specify the amount of memory installed both on the System Board and in the System Expansion Slots. Refer to the System Board Component Diagram (page 2-13) for DIP switch locations.

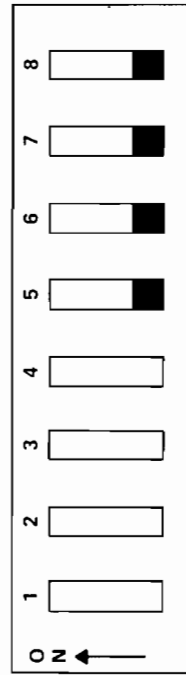
SWITCH 1



Position Function

- 1-7-8 Number of 5-1/4" Diskette Drives Installed; page 2-29
- 2 Unused — must be on (reserved for co-processor)
- 3-4 Amount of memory on System Board; pages 2-30 to 2-31
- 5-6 Type of monitor you are using; page 2-29

SWITCH 2

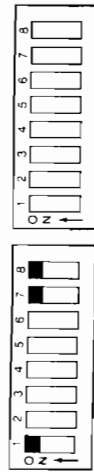


Position Function

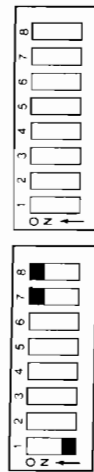
- 1-2-3-4 Amount of memory options installed; page 2-30 to 2-31
- 5-6-7-8 Always in the OFF position

Number of 5-1/4" Diskette Drives Installed

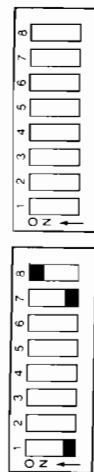
SWITCH 1 SWITCH 2



0-Drives



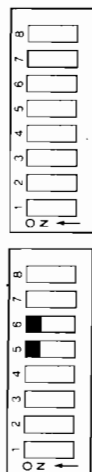
1-Drive



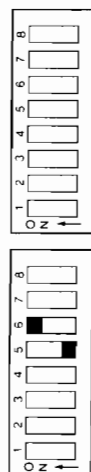
2-Drives

Monitor Type

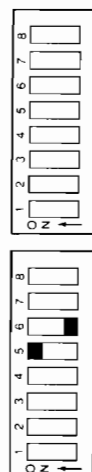
SWITCH 1 SWITCH 2



No Monitors

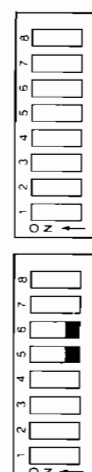


40 x 25 Color



80 x 25 Color

Note: The 80 x 25 color setting, when used with home television and various monitors can cause loss of character/quality.



IBM Monochrome Display or more than one monitor

System Board Memory Switch Settings

| Total Memory | Switch 1 | Switch 2 |
|--------------|----------|----------|
| 16KB | | |
| 32KB | | |
| 48KB | | |
| 64KB | | |
| 96KB | | |
| 128KB | | |
| 160KB | | |
| 192KB | | |
| 224KB | | |
| 256KB | | |

SYSTEM BOARD

MEMORY OPTIONS


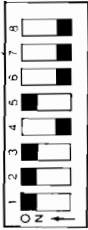

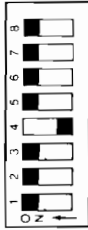

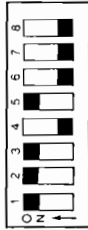
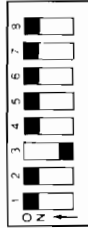
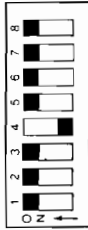
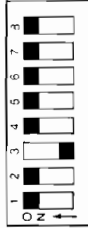
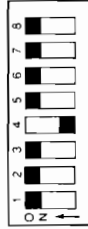
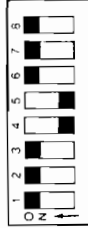
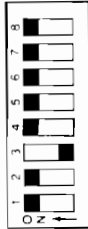
System Board Memory Switch Settings

| Total Memory | Switch 1 | Switch 2 |
|--------------|----------|----------|
| 288KB | | |
| 320KB | | |
| 352KB | | |
| 384KB | | |
| 416KB | | |
| 448KB | | |
| 480KB | | |
| 512KB | | |
| 544KB | | |

MEMORY OPTIONS

Memory Expansion Options


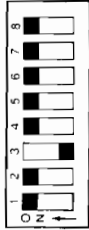









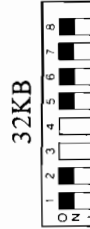
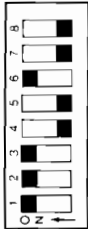

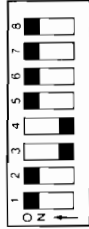





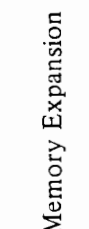
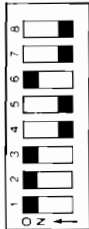


The following are switch settings on each Memory Expansion Option.

| Total Memory | Option 1 | Option 2 |
|--------------|---|---|
| 96KB |  | |
| 128KB |  | |
| 128KB |  | |
| 128KB |  |  |
| 160KB |  |  |
| 160KB |  |  |
| 160KB |  |  |
| | |  |

*Amount of memory installed on a 64/256KB Memory Expansion Option.

Memory Expansion Options

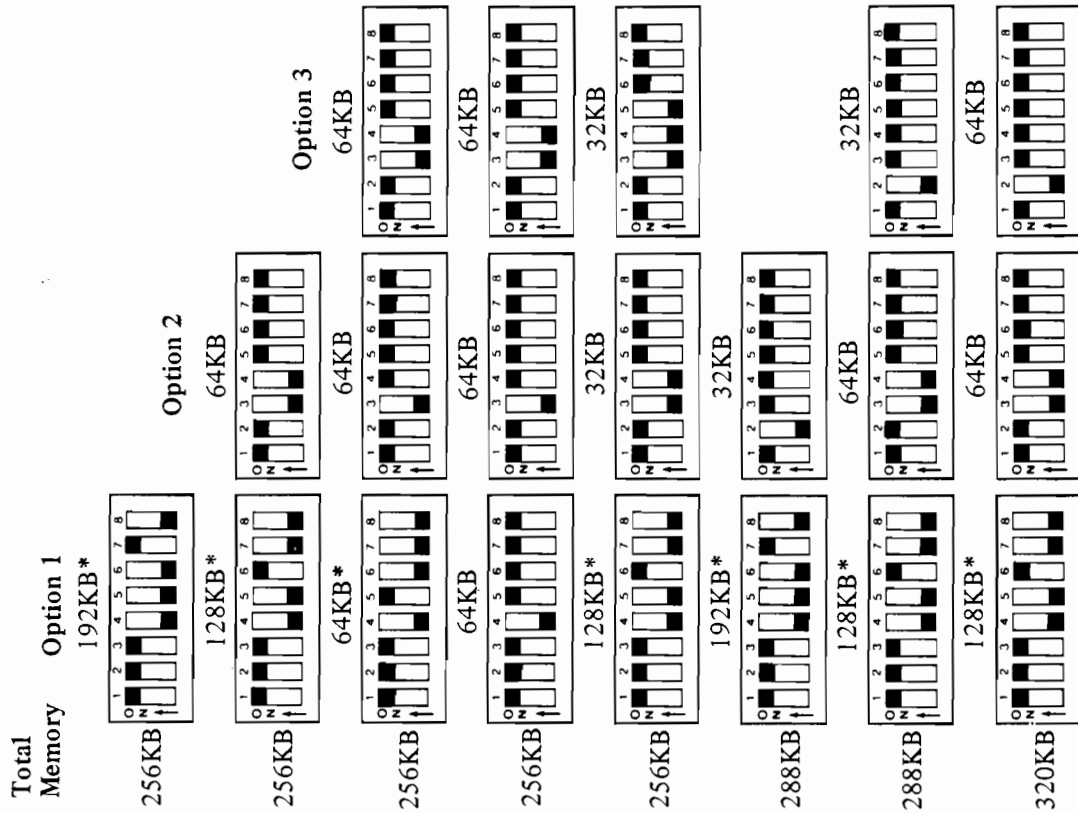
The following are switch settings on each Memory Expansion Option.

| Total Memory | Option 1 | Option 2 | Option 3 |
|--------------|---|---|--|
| 192KB |  |  |  |
| 192KB |  |  |  |
| 192KB |  |  |  |
| 192KB |  |  |  |
| 192KB |  |  |  |
| 224KB |  |  |  |
| 224KB |  |  |  |
| 224KB |  |  |  |

*Amount of memory installed on a 64/256KB Memory Expansion Option.

Memory Expansion Options

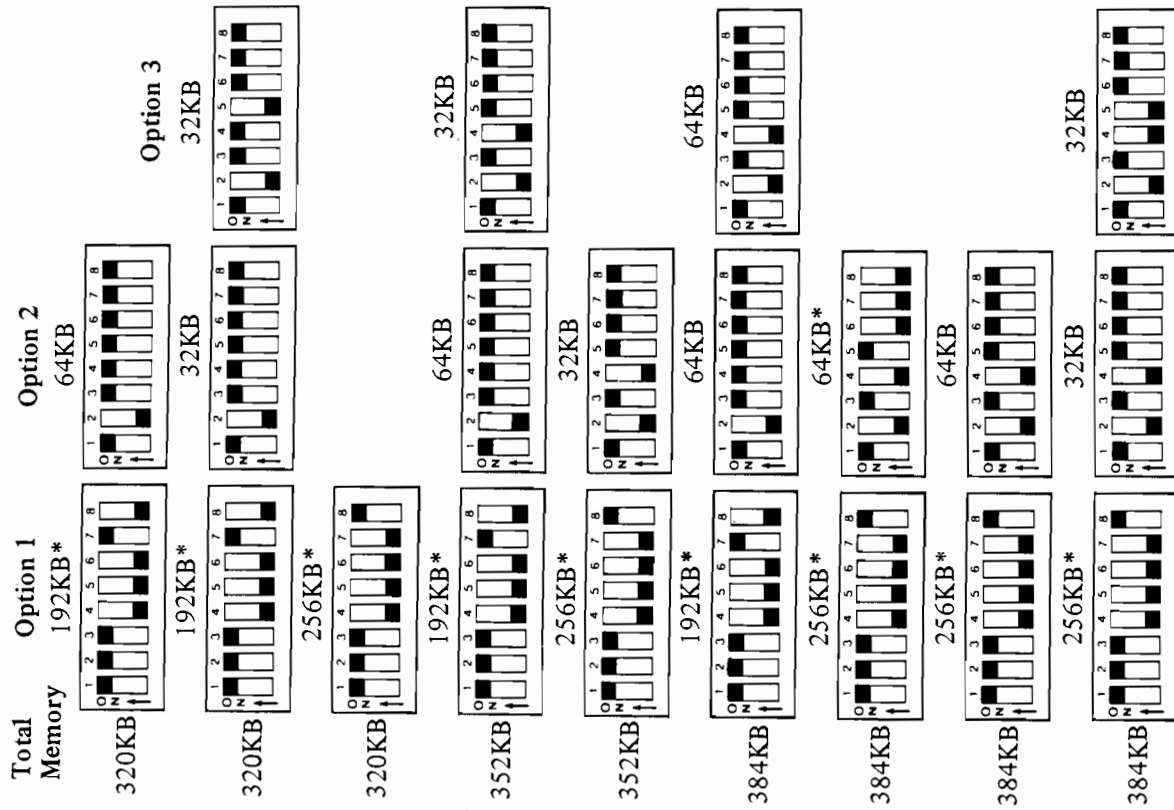
The following are switch settings on each Memory Expansion Option.



*Amount of memory installed on a 64/256KB Memory Expansion Option.

Memory Expansion Options

The following are switch settings on each Memory Expansion Option.



*Amount of memory installed on a 64/256KB Memory Expansion Option.

Memory Expansion Options

The following are switch settings on each Memory Expansion Option.

| Total Memory | Option 1 256KB* | Option 2 64KB* | Option 3 32KB |
|--------------|--------------------|-------------------|------------------|
| 416KB | | | |
| 416KB | | | |
| 448KB | | | |
| 448KB | | | |
| 448KB | | | |
| 480KB | | | |
| 512KB | | | |
| 512KB | | | |
| 544KB | | | |

*Amount of memory installed on a 64/256KB Memory Expansion Option.

Power Supply

The system DC power supply is a 63.5 watt, 4 voltage level switching regulator. It is integrated into the System Unit and supplies power for the System Unit, its options, and the keyboard. The supply provides 7 amps of +5 Vdc, $\pm 5\%$ 2 amps of +12 Vdc, $\pm 5\%$ 300 ma of -5 Vdc, $\pm 10\%$ and 250 ma of -12 Vdc, $\pm 10\%$. All power levels are regulated with overvoltage and over current protection. If DC over-load or over-voltage conditions exist, the supply will automatically shut down until the condition is corrected. The supply is designed for continuous operation at 63.5 watts and the input to the supply is fused.

There are two different power supplies available for the system unit: one that has an input voltage of 120 Vac and one that has an input voltage of 220/240 Vac. Other than their input ratings the two power supplies have identical specifications.

The System Board takes approximately 3 amps of +5 Vdc thus allowing approximately 4 amps of 5 Vdc for the adapters in the System Expansion Slots. The +12 Vdc power level is designed to power the two internal 5-1/4" Diskette Drives and the system's dynamic memory. It is assumed that only one drive motor is active at a time. The -5 Vdc level is used for memory bias voltage and analog circuits in the diskette adapter phase lock loop. The +12 Vdc and -12 Vdc are used for powering the serial interface card EIA drivers and receivers for the Asynchronous Communications Adapter. All four power levels are bussed across the five System Expansion Slots and available for option adapter.

The IBM Monochrome Display has its own power supply. This high resolution display receives its AC power from the System Unit power supply and is switched on and off with the power switch, which saves a wall outlet. The AC output for the display is a nonstandard connector, so only the AC high resolution Display can use this AC port.

Power Supply Location

The Power Supply is located at the right rear area of the System Unit. It supplies operating voltages to the System Board, IBM Monochrome Display, and provides two separate connections for power to the 5-1/4" Diskette Drives (if installed). The nominal power requirements and output voltages are listed on the following tables:

Input Requirements 120 Volt

| VOLTAGE Vac | | FREQUENCY Hz | CURRENT MAX |
|----------------|---------|-----------------|--------------------------|
| NOMINAL | MINIMUM | MAXIMUM | @V _{in} MINIMUM |
| 120 | 104 | ± 3 Hz 60 | 2.5 AMPS @ 104 Vac |

Input Requirements 220/240 Volt

| VOLTAGE Vac | | FREQUENCY Hz | CURRENT MAX |
|----------------|---------|-----------------|--------------------------|
| NOMINAL | MINIMUM | MAXIMUM | @V _{in} MINIMUM |
| 220/240 | 180 | ± 3 Hz 50 | 1.0 AMPS @ 180 Vac |

DC Output

| VOLTAGE V _{dc} | CURRENT AMPS | | REGULATION TOLERANCE | |
|----------------------------|-----------------|------|-------------------------|-----|
| NOMINAL | MIN | MAX | ± % | — % |
| + 5.0 | 2.3 | 7.0 | 5 | 4 |
| — 5.0 | 0.0 | 0.3 | 10 | 8 |
| +12.0 | 0.0 | 2.0 | 5 | 4 |
| —12.0 | 0.0 | 0.25 | 10 | 9 |

AC Output

The AC voltage out is connected to the input voltage through the power supply fuse filter card and On/Off switch. Since this output is designed to support only the IBM Monochrome Display and since the supply input is fused, there is only a limited amount of current available: .75 A maximum on the 120 Vac supply and .38 A maximum on the 220/240 Vac power supply.

Power Supply Connectors and Pin Assignment

The power connector on the System Board is a 12 pin male connector which plugs into the power supply connectors. The pin configurations and locations are shown below:

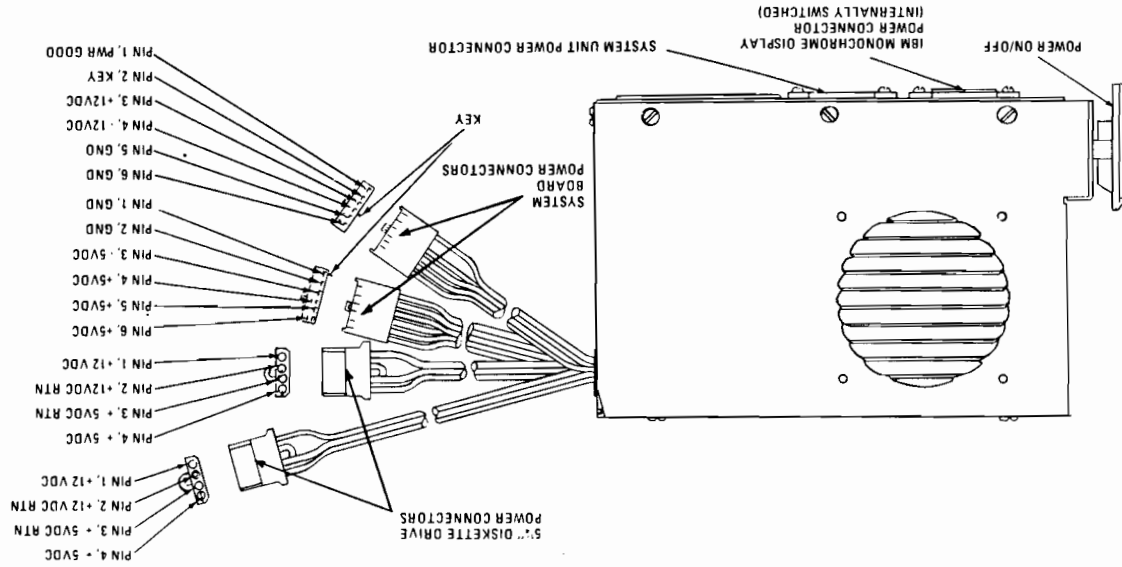


Figure 13. POWER SUPPLY AND CONNECTORS

IBM Monochrome Display and Parallel Printer Adapter

Important Operating Characteristics

Over Voltage/Current Protection

PRIMARY (INPUT)

| VOLTAGE NOMINAL VAC | TYPE PROTECTION | RATING AMPS |
|---------------------------|--|-----------------|
| 120 240 | 60 Hz FUSE TYPE 2 SOC SD4 50 Hz FUSE TYPE 2 SOC SD4 | 2 AMPS 1 AMP |

Power On/Off Cycle: When the supply is turned off for a minimum of 5 seconds, and then turned on, the power good signal will be regenerated.

Signal Requirements

The power good signal indicates that there is adequate power to continue processing. If the power goes below the specified levels, the power good signal triggers a system shut-down.

The Power Supply provides a power good signal out, to indicate that the +/-5V and +/-12V outputs are above the sense level defined in the chart below. The power good signal is up level (2.4V to 5.5V), TTL compatible, and capable of sourcing 60 U.A. When any of the four sensed output voltages is below its sense level voltage as defined in the chart below, the power good signal is down level (0V to 0.4V), TTL compatible, and capable of sinking 500 U.A. The power good signal (after all levels of the output voltage are good) has a turn on delay of at least 100 MS but no more than 500 MS.

The sense levels of the +/-5V and +/-12V outputs are:

| OUTPUT | MIN | SENSE VOLTAGE NOMINAL | MAX |
|--------|------|--------------------------|-------|
| +5V | +3.7 | +4.0 | +4.3 |
| -5V | -3.7 | -4.0 | -4.3 |
| +12V | +8.5 | +9.6 | +10.5 |
| -12V | -8.5 | -9.6 | -10.5 |

This adapter has dual functions. The first is to provide the interface to the IBM Monochrome Display. The second function is a parallel interface for the IBM 80 CPS Matrix Printer.

The monitor interface is designed around the Motorola 6845 CRT Controller module. There are 4K bytes of static memory on the card which are used for the display buffer. The memory is dual ported and may be accessed directly by the CPU. No parity is provided on the display buffer. A block diagram of the Monochrome Display function in on page 2-48.

The characteristics of the design are listed below:

- 80x25 Screen
- Direct Drive Output
- 9x14 Character Box
- 7x9 Character
- 18 Khz Monitor
- Character Attributes

The adapter supports 256 character codes. An 8K byte character generator contains the fonts for the character codes. The characters, values, keystrokes and screen characteristics are tabled in Appendix C. Of Characters, Keystrokes and Color.

Note: This Adapter when used with a display containing P39 Phosphor, will not support a light pen!

Parallel Interface Description

This topic is discussed in full on pages 2-75 through 2-79.

IBM Monochrome Display Adapter Block Diagram

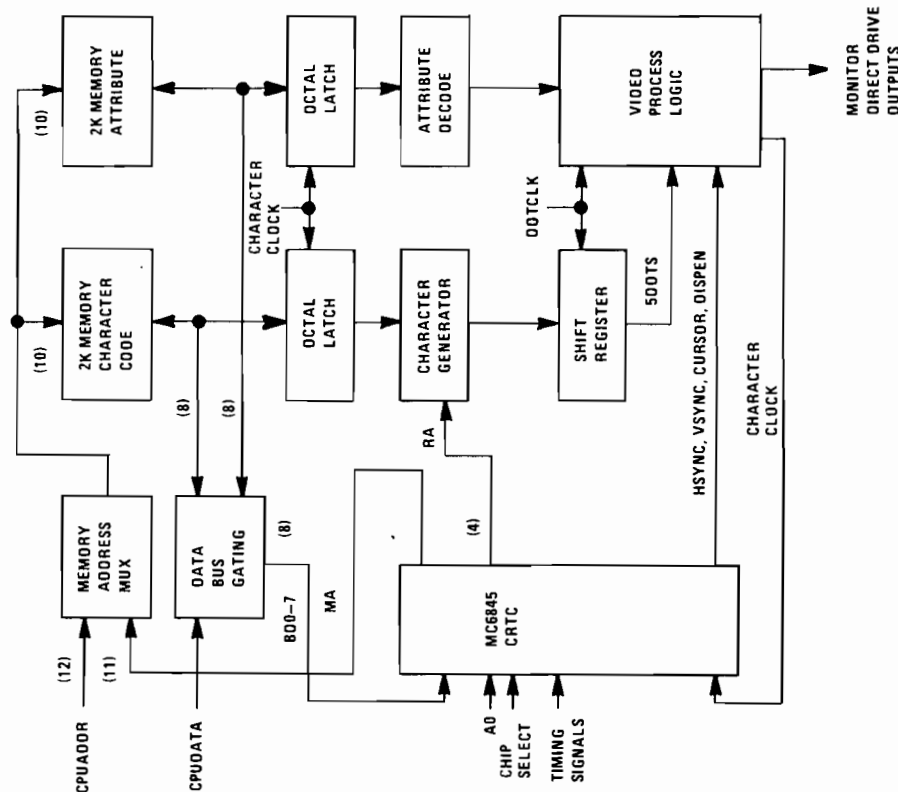


Figure 14. IBM MONOCHROME DISPLAY ADAPTER BLOCK DIAGRAM

System Channel Interface

Lines Used

This card uses the address and data bus, memory and I/O read/write signals, reset, I/O Ready, I/O Clock, and IRQ7.

Loads

Where possible, only one "LS" load is on the signals present at the I/O slot. Some of the address bus lines have two "LS" loads. No signal has more than two "LS" loads.

Special Timing

At least one wait state will be inserted on all memory and I/O accesses from the CPU. The duration of the wait-state will vary because the CPU/monitor access is synchronized with the character clock on this adapter.

To insure proper initialization of the attachment, the first instruction issued to the card must be to set the high resolution bit of the monitor output Port 1. (OUT PORT 3B8 = 01H). A CPU access to this adapter must never occur if the high resolution bit is not set.

System configurations which have two display adapter cards must insure that both adapters are properly initialized after a power on reset. Damage to either display may occur if not properly initialized.

Data Rates

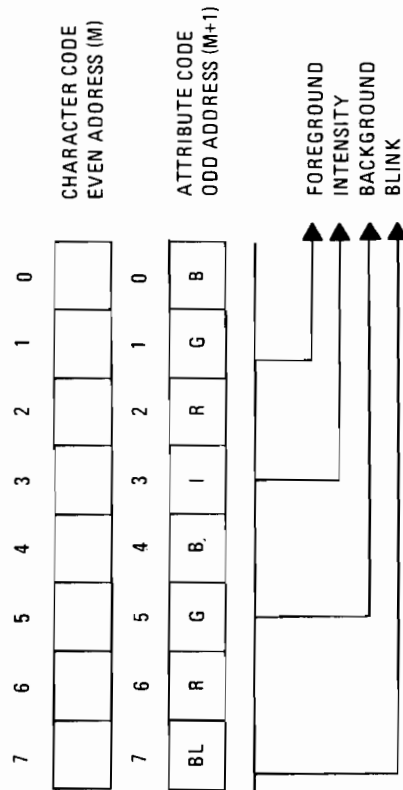
For the IBM Monochrome Display Adapter, two bytes are fetched from the display buffer in 553 ns providing a data rate of 1.8M bytes/second.

Interrupt and DMA Response Requirements

- The display buffer can be written into, or read from using DMA.
- The parallel interface uses the +IRQ7 line. Interrupt becomes active when the acknowledge input is low, and interrupts are enabled via the control port.

Modes of Operation

The IBM Monochrome Display and Printer Adapter supports 256 character codes. In the character set are alphanumerics and block graphics. Each character in the display buffer has a corresponding character attribute. The character code must be an even address and the attribute code must be an odd address in the display buffer.



The adapter decodes the character attribute byte as defined above. The **BLINK** and **INTENSITY** bits may be combined with the **FOREGROUND** and **BACKGROUND** bits to further enhance the character attribute functions listed below.

| BACKGROUND R G B | FOREGROUND R G B | FUNCTION |
|---------------------|---------------------|--------------------------------------|
| 0 0 0 | 0 0 0 | NON DISPLAY |
| 0 0 0 | 0 0 1 | UNDERLINE |
| 0 0 0 | 1 1 1 | WHITE CHARACTER/ BLACK BACKGROUND |
| 1 1 1 | 0 0 0 | REVERSE VIDEO |

Programming Considerations

Programming the 6845 CRT Controller

The following table summarizes the 6845 Internal Data Registers and their functions and parameters. For the IBM Monochrome Display, the values in the table must be programmed into the 6845 to insure proper initialization of the device.

Table 2. 6845 Initialization Parameters

| REGISTER = | REGISTER FILE | PROGRAM UNIT | 80x25 MONOCHROME |
|---------------|-----------------------|-----------------|---------------------|
| R0 | HORIZONTAL TOTAL | CHARACTERS | 61H |
| R1 | HORIZONTAL DISPLAYED | CHARACTERS | 50H |
| R2 | HSYNC POSITION | CHARACTERS | 52H |
| R3 | HSYNC WIDTH | CHARACTERS | FH |
| R4 | VERTICAL TOTAL | CHAR ROWS | 19H |
| R5 | VERTICAL ADJUST | SCAN LINE | 6H |
| R6 | VERTICAL DISPLAYED | CHAR ROW | 19H |
| R7 | VSYNC POSITION | CHAR ROW | 19H |
| R8 | INTERLACE MODE | --- | 02 |
| R9 | MAX SCAN LINE ADDRESS | SCAN LINE | DH |
| R10 | CURSOR START | SCAN LINE | 8H |
| R11 | CURSOR END | SCAN LINE | CH |
| R12 | START ADDRESS (H) | --- | 00H |
| R13 | START ADDRESS (L) | --- | 00H |
| R14 | CURSOR (H) | --- | 00H |
| R15 | CURSOR (L) | --- | 00H |
| R16 | RESERVED | --- | --- |
| R17 | RESERVED | --- | --- |

Sequence of Events

The first command issued to this attachment must be to output to PORT 3B8, hex 01, to set high resolution mode. If the high resolution mode is not set, an infinite CPU wait-state will occur!

Memory Requirements

The attachment has 4K bytes of memory which is used for the display buffer. The memory supports one screen of 25 rows of 80 characters, plus a character attribute for each display character. No parity is provided on the memory. No system Read/Write memory is required for the monochrome adapter portion. The display buffer starts at address 'B0000'.

DMA Channels

The display buffer will support a DMA operation, however CPU wait-states will be inserted during DMA.

Interrupt Levels

Interrupt Level 7 is used on the parallel interface. Interrupts can be enabled or disabled via the Printer Control Port. The interrupt is a high level active signal.

I/O Address and Bit Map

The table below breaks down the functions of the I/O Address decode for the card. The I/O address decode is from '3B0' through '3BF'. The bit assignment for each I/O address follows:

I/O Address Function

| | |
|-----|----------------------|
| 3B0 | Not Used |
| 3B1 | Not Used |
| 3B2 | Not Used |
| 3B3 | Not Used |
| 3B4 | 6845 Index Register |
| 3B5 | 6845 Data Register |
| 3B6 | Not Used |
| 3B7 | Not Used |
| 3B8 | CRT Control Port 1 |
| 3B9 | Reserved |
| 3BA | CRT Status Port |
| 3BB | Reserved |
| 3BC | Parallel Data Port |
| 3BD | Printer Status Port |
| 3BE | Printer Control Port |
| 3BF | Not Used |

The 6845 Index and Data Registers are used to program the CRT controller to interface to the high resolution Monochrome Display.

- CRT Output Port 1 (I/O Address '3B8')

| Bit # | Function |
|-------|------------------------|
| 0 | + high resolution mode |
| 1 | Not Used |
| 2 | Not Used |
| 3 | + video enable |
| 4 | Not Used |
| 5 | + enable blink |
| 6, 7 | Not Used |

- CRT Status Port (I/O Address '3BA')

| Bit # | Function |
|-------|--------------------|
| 0 | + horizontal Drive |
| 1 | Reserved |
| 2 | Reserved |
| 3 | + B/W Video |

IBM Monochrome Display

The high resolution IBM Monochrome Display unit attaches to the System Unit via two cables of approximately 3' (914 mm) in length. One cable is a signal cable which contains direct drive interface from the IBM Monochrome Display and Printer Adapter.

The second cable provides AC power to the display from the System Unit. This allows the System Unit power On/Off switch to also control the display unit. An additional benefit is a reduction in the requirements for wall outlets to power the system. The monitor contains an 11.5" (283.1 mm) diagonal 90° deflection CRT. The CRT and analog circuits are packaged in an enclosure so the display may either sit on top of the System Unit or on a nearby table top or desk. The unit has both brightness and contrast adjustment controls on the front available to the operator.

Operating Characteristics

Screen

High persistence green phosphor (P 39) with an etched surface to reduce glare. Unit displays an 80 character by 25 line screen with a 9 dot wide by 14 dot tall character box.

Video Signal

Maximum video bandwidth of 16.27 Mhz.

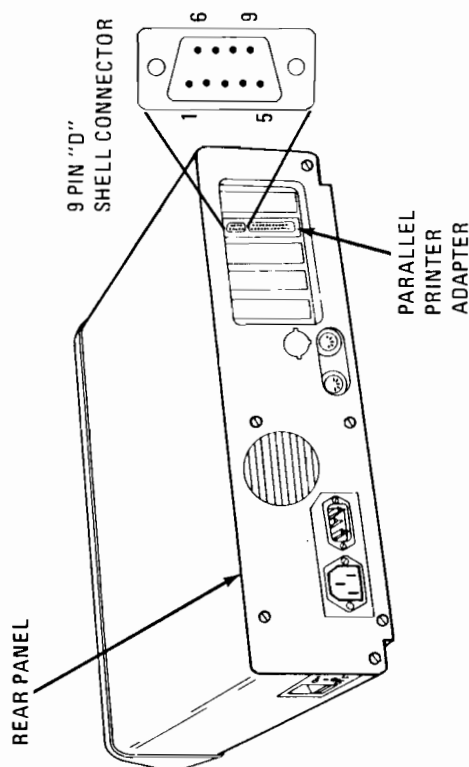
Vertical Drive

Screen refreshed at 50 Hz with 350 vertical lines of resolution and 720 lines of horizontal resolution measured at center of screen.

Horizontal Drive

Positive level TTL compatible frequency, 18.432 KHz.

IBM Monochrome Direct Drive Interface and Pin Assignment



At Standard TTL Levels

| IBM Monochrome Display | At Standard TTL Levels | | | | | | | | | IBM Monochrome Display and Parallel Printer Adapter |
|------------------------|------------------------|--------|----------|----------|----------|-------------|---------|--------------|------------|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | |
| | Ground | Ground | Not Used | Not Used | Not Used | + Intensity | + Video | + Horizontal | - Vertical | |

NOTE: Signal voltages are 0 - .6 Vdc at down level
+5 Vdc at high level

Color/Graphics Monitor Adapter

The Color/Graphics Monitor Adapter is designed to attach a wide variety of NTSC (U.S. National Television System Committee) standard TV frequency monitors and TV sets (user-supplied RF modulator required for TVs). It is capable of operating in black and white or color, and provides three video interfaces: a composite video port, a direct drive port, and a connection interface for driving a user supplied RF modulator. In addition, a light pen interface is provided.

The adapter has two basic modes of operation; alphanumeric (A/N) and all points addressable graphics (APA). Additional modes are available within A/N and APA modes. In A/N mode, the display can be operated in a 40x25 mode for low resolution monitors and TVs or 80x25 mode for high resolution monitors. In both modes, characters are defined in an 8x8 box and are 7x7 with one line of descender for lowercase (both uppercase and lowercase characters are supported in all modes). In black and white mode, the character attributes of Reverse Video, Blinking and High-lighting are available. In color mode, there are 16 foreground colors and 8 background colors available per character. In addition, blinking on a per character basis is available.

The adapter card contains 16KB of storage; thus, for a 40x25 screen, 1000 bytes are used to store character information and 1000 bytes are used for attribute/color information. This means that up to 8 pages of screens can be stored in the adapter memory. Similarly, in an 80x25 mode, 4 pages of display screen may be stored in the adapter. The full 16KB storage on the display adapter is directly addressable by the processor allowing maximum software flexibility in managing the screen. In A/N color modes, it is also possible to select the screen border color. One of 16 colors may be selected.

In APA mode, there are two resolutions available; 320x200 and 640x200. In the 320x200, each (picture element) pel may have one of four colors. The background color (color 0) may be any of the 16 possible colors. The remaining 3 colors come from one of the two software selectable palettes. One palette contains red/green/brown, the other contains cyan/magenta/white.

The 640x200 mode is only available in black and white since the full 16KB of storage is used to define the on or off state of the pel. The adapter operates in noninterlace mode at either 7 or 14 megahertz (Mhz) video bandwidth depending on the mode of operation selected. Vertical scan output frequency is 60 Hz and horizontal output frequency is 15,750 Hz.

In A/N mode, characters are formed from a ROM character generator. The character generator contains dot patterns for 256 characters. The character set contains the following major grouping of characters. Sixteen special characters for game support, 15 characters for support of word processing editing functions, the standard 96 ASCII graphic set, 48 characters to support foreign languages, 48 characters for business block graphics allowing drawing of charts, boxes and tables using single and double lines, 16 of the most often used Greek characters, and 15 of the most often used scientific notation characters.

The Color/Graphics Monitor Adapter Function is packaged on a single card which fits into one of the five System Expansions Slots on the System Board. The direct drive and composite video ports are right-angle mounted connectors at the rear of the adapter and extend through the rear panel of the System Unit.

The display adapter is implemented using a Motorola 6845 CRT controller device. This adapter is highly programmable with respect to raster and character parameters. A block diagram of the Color/Graphics Adapter is on the following page.

Color/Graphics Monitor Adapter Block Diagram

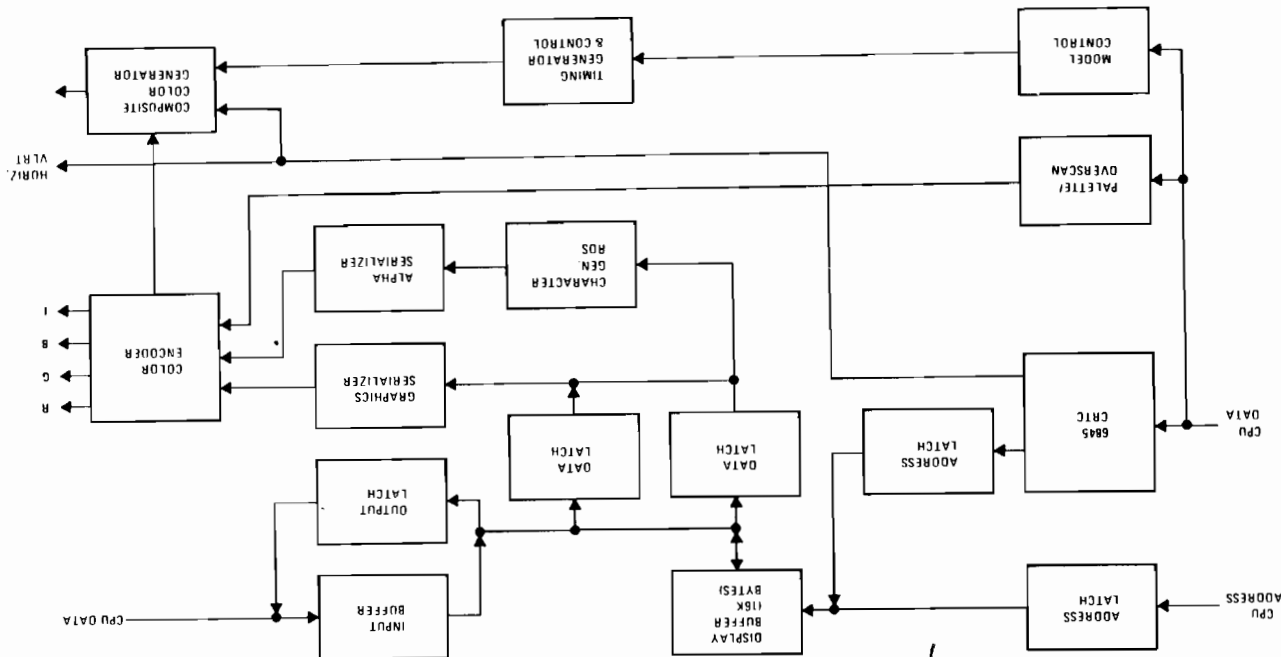


Figure 15. COLOR/GRAPHICS MONITOR ADAPTER BLOCK DIAGRAM

Major Components Definitions

Motorola 6845 CRT Controller

This device provides the necessary interface to drive a raster scan CRT.

Mode Set and Status Registers

This is a general purpose programmable I/O register. It has I/O points which may be individually programmed. Its function in this attachment is to provide mode selection (page 2-49 and 2-50) and color selection in the medium resolution color graphics mode (page 2-51).

Display Buffer

The Display Buffer resides in the CPU address space starting at address X'B8000'. It provides 16K bytes of dynamic read/write memory. A dual-ported implementation allows the CPU and the graphics control unit to access this buffer. The CPU and the CRT control unit have equal access to this buffer during all modes of operation except in high resolution alphanumeric mode. In this mode the CPU should access this buffer during the horizontal retrace intervals. The CPU may however, write to the required buffer at any time, but a small amount of display fetches will result if not during retrace intervals.

Character Generator

This attachment utilizes a ROM character generator. It consists of 8K bytes of storage which cannot be read/written under software control. This is a general purpose ROM character generator with three different character fonts. Two character fonts are used on this card (a 7x7 double dot and 5x7 single dot), selected by a card jumper. No jumper gives a 7x7 double dot; with a jumper a single dot font is selected.

Timing Generator

This block generates the timing signals used by the 6845 CRT controller and by the dynamic memory. It also resolves the CPU/graphic controller contentions for accessing the Display Buffer.

Composite Color Generator

The logic in this block generates base band video color information.

Modes of Operation

There are two basic modes of operation, 'Alphanumeric' and 'Graphics'. Each of these modes provide further options in both color and black-and-white. The following text describes each mode of operation.

Alphanumeric Mode

Alphanumeric Display Architecture

Every display character position is defined by two bytes in the regen buffer (part of display adapter, not system memory). Both the color and the black and white display adapter use this 2 byte character/attribute format.

| DISPLAY CHAR CODE BYTE | | | | | | | | ATTRIBUTE BYTE | | | | | | | |
|------------------------|---|---|---|---|---|---|---|----------------|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | | | | | | | | | | | |

Attribute Byte Definition

| ATTRIBUTE BYTE | | | | | | | | |
|---------------------|----|------------|---|---|---|------------|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ATTRIBUTE FUNCTION | B | R | G | B | I | R | G | B |
| | FG | BACKGROUND | | | | FOREGROUND | | |
| NORMAL | B | 0 | 0 | 0 | I | 1 | 1 | 1 |
| REVERSE VIDEO | B | 1 | 1 | 1 | I | 0 | 0 | 0 |
| NON DISPLAY (BLK) | B | 0 | 0 | 0 | I | 0 | 0 | 0 |
| NON DISPLAY (WHITE) | B | 1 | 1 | 1 | I | 1 | 1 | 1 |

| = HIGH LIGHT FOREGROUND (CHAR)

B=BLINK FOREGROUND (CHAR)

Color TV (NTSC - U.S. Standard)

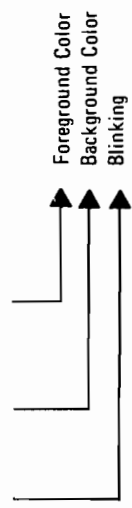
- Display up to 25 rows of 40 characters each
- Maximum of 256 characters
- Requires 2000 bytes of Read/Write Memory (on the adapter)
- 8x8 character box
- 7x7 double dotted characters (one descender)
- Character attributes (one for each character)

| | | | | | | | | | | | | | | | |
|------------------------------------|---|---|---|---|---|---|---|----------------|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CHARACTER CODE | | | | | | | | ATTRIBUTE CODE | | | | | | | |
| EVEN ADDRESS (M) ODD ADDRESS (M+1) | | | | | | | | | | | | | | | |

ATTRIBUTE BYTE DEFINITIONS

R: Red
G: Green
B: Blue
I: Intensity

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 8 | R | G | B | I | R | G | B |



Note: The starting address of the display buffer must be an even location.

Color Monitor (with Drive input capability and 60 Hz vertical refresh)

Display up to 25 rows of 80 characters each

Requires 4000 bytes of Read/Write Memory (on the adapter)

Maximum of 256 character set

8x8 character box

7x7 character with one descender

Same format for attributes as for color TV

Note: The starting address of the display buffer must be an even location.

IBM Monochrome Display Adapter Vs. Color/Graphics Adapter Attribute Relationship

Table 3. Monochrome Vs. Color/Graphics Attributes

| NORMAL RVV NON DISP (BLK) NON DISP (WHT) | ON THE MONOCHROME DISPLAY ADAPTER | | | | | | | | ON THE COLOR/GRAPHIC DISPLAY ADAPTER | | | |
|---|---|---|---|---|---|---|---|---|--|-------------|-------------|-------------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | CHAR. COLOR | 8KGD. COLOR | CHAR. COLOR | 8KGD. COLOR |
| | FG BACKGROUND FOREGROUND | | | | | | | | | | | |
| | 8 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | WHITE | BLACK | WHITE | BLACK |
| | 8 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | BLACK | WHITE | BLACK | WHITE |
| 8 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | BLACK | BLACK | BLACK | |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | WHITE | WHITE | WHITE | |
| | ALL OTHER CODES DEFINE FOREGROUND COMBINATIONS | | | | | | | | ALL OTHER CODES CHANGE FOREGROUND BACKGROUND COLOR TO SELECTED VALUE | | | |

ALL OTHER CODES RESULT IN WHITE BACKGROUND
CHAR ON BLACK BACKGROUND
ALL OTHER CODES CHANGE FOREGROUND BACKGROUND COLOR TO SELECTED VALUE

R G B
0 0 0 = BLACK
0 0 1 = BLUE
0 1 0 = GREEN
0 1 1 = CYAN
1 0 0 = RED
1 0 1 = MAGENTA
1 1 0 = YELLOW
1 1 1 = WHITE

AN ADDITIONAL 8 COLOR (ACTUAL) DIFFERENT SHADES OF THE ABOVE) ARE SELECTED BY SETTING THE (I) BIT

Note: Not all Monitors Recognize the (I) Bit

Table 4. Color/Graphics Modes

| | HORIZONTAL | VERTICAL | NO OF COLORS (INCL. BACKGROUND COLOR) |
|----------|------------|----------|--|
| LOW RES | 160 | 100 | 16 (INCLUDES BLACK AND WHITE) |
| MED RES | 320 | 200 | 4 COLORS: 1 OF 16 FOR BACKGROUND PLUS GREEN, RED, YELLOW OR CYAN, MAGENTA, WHITE |
| HIGH RES | 640 | 200 | 8 B & W ONLY |

1. Low resolution color graphics (TV or monitor).

(Note: This mode is not supported in ROM).

- Up to 100 rows of 160 pels each (2x2)
 - 1 of 16 colors each pel specified by I, R, G and B
 - Requires 8000 byte of Read/Write Memory (on the adapter)
 - Memory mapped graphics (requires special memory map and set up to be defined later)
2. Medium resolution color graphics (TV or monitor)
 - Up to 200 rows of 320 pels each (1x1)
 - 1 out of 4 preselected colors in each box
 - Requires 16000 bytes of Read/Write Memory (on the adapter)

- Memory mapped graphics
- 4 pels/byte

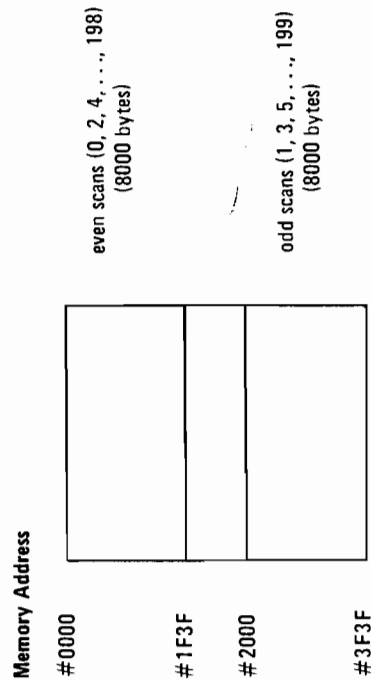
FORMAT:

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| C1 | C0 | C1 | C0 | C1 | C0 | C1 | C0 |

First display
pel

- Graphics storage is organized in two banks of 8000 bytes each.

Graphics Storage Map



Address #0000 contains pel information for upper left corner of display area.

Color selection is determined by the following logic:
C1 and C0 will select 4 of 16 preselected colors.

This color selection (palette) is preloaded in an I/O port.

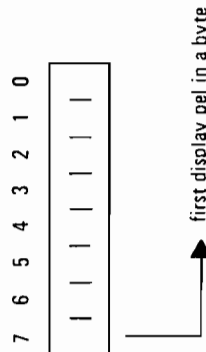
| C1 | C0 | CODE SELECT COLOR FOR DISPLAY POSITION |
|----|----|--|
| 0 | 0 | DOT TAKES ON COLOR OF 1 OF 16 PRESELECTED BACKGROUND COLORS. |
| 0 | 1 | SELECT 1ST COLOR OF PRESELECT COLOR SET "1" OR "2" |
| 1 | 0 | SELECT 2ND COLOR OF PRESELECT COLOR SET "1" OR "2" |
| 1 | 1 | SELECT 3RD COLOR OF PRESELECT COLOR SET "1" OR "2" |

The two color sets are:

| SET ONE | SET TWO |
|-------------------|-----------------|
| COLOR 1 - CYAN | COLOR 1 - GREEN |
| COLOR 2 - MAGENTA | COLOR 2 - RED |
| COLOR 3 - WHITE | COLOR 3 - BROWN |

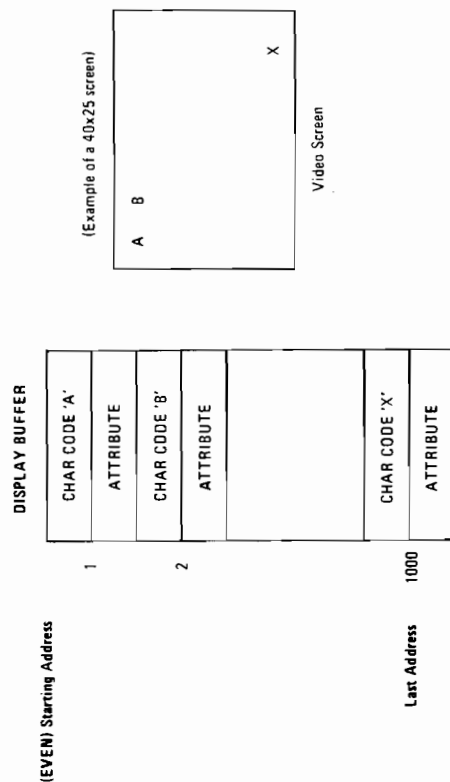
The background colors are the same basic 8 color as defined for low resolution graphic plus 8 alternate intensities defined by the intensity bit for a total of 16 color including black and white.

3. Black and white high resolution graphics (monitor)
 - Up to 200 rows of 640 pels each (1x1)
 - Black and white only
 - Requires 16000 bytes of Read/Write Memory (on the adapter)
 - Addressing and mapping is the same as for medium resolution color graphics, but the data format is different. In this mode each bit in memory is mapped to a pel on the screen.
 - 8 pels/byte



Description of Basic Operations

In the alphanumeric mode the adapter fetches character and attribute information from its display buffer. The starting address of the display buffer is programmable through the 6845, but it must be an even address. The character codes and attributes are then displayed according to their relative position in the buffer.



The CPU and the display control unit have equal access to the display buffer during all the operating modes except high resolution alphanumeric. During this mode, the CPU should access the display buffer during the vertical retrace time (if not, then the display will be affected with random patterns as the CPU is using the display buffer). The characters are displayed from a prestored "character generator" which contains the dot patterns of all the displayable characters.

In the graphics mode the displayed dots and colors are also fetched from the display buffer (up to 16K bytes). In the Color/Graphics Mode Section, the bit configuration for each graphics mode is explained.

Table 5. Summary of Available Colors

| I | R | G | B | COLOR |
|---|---|---|---|---------------|
| 0 | 0 | 0 | 0 | Black |
| 0 | 0 | 0 | 1 | Blue |
| 0 | 0 | 1 | 0 | Green |
| 0 | 0 | 1 | 1 | Cyan |
| 0 | 1 | 0 | 0 | Red |
| 0 | 1 | 0 | 1 | Magenta |
| 0 | 1 | 1 | 0 | Brown |
| 0 | 1 | 1 | 1 | Light Gray |
| 1 | 0 | 0 | 0 | Dark Gray |
| 1 | 0 | 0 | 1 | Light Blue |
| 1 | 0 | 1 | 0 | Light Green |
| 1 | 0 | 1 | 1 | Light Cyan |
| 1 | 1 | 0 | 0 | Light Red |
| 1 | 1 | 0 | 1 | Light Magenta |
| 1 | 1 | 1 | 0 | Yellow |
| 1 | 1 | 1 | 1 | White |

Note: "I" provides extra luminance (brightness) to each shade available. Resulting in the light colors listed above, except where the "I" bit is not recognized by some monitors.

Programming Considerations

Programming the 6845 CRT Controller

The 6845 has 19 internal registers which are used to define and control a raster scanned CRT display. One of these registers, the Address Register, is actually used as a pointer to the other 18 registers. It is a write only register which is loaded from the CPU by executing an OUT instruction to I/O address 3D4. The five least significant bits of the I/O bus are loaded into the Address Register.

In order to load any of the other 18 registers, the Address Register is first loaded with the necessary pointer and then the CPU may output a value to I/O address 3D5 in order to load the information in the preselected register.

The following table defines the values which must be loaded in 6845 Registers in order to control the different modes of operation supported by the attachment.

Table 6. 6845 Register Description

| ADDR REG. | REG. # | REGISTER TYPE | UNITS | I/O | 40x25 ALPHA | 80x25 ALPHA | GRAPHIC MODES |
|-----------|--------|-----------------------|-----------|------------|-------------|-------------|---------------|
| 0 | R0 | Horizontal Total | Char. | Write Only | 38 | 71 | 38 |
| 1 | R1 | Horizontal Displayed | Char. | Write Only | 28 | 50 | 28 |
| 2 | R2 | Horiz. Sync Position | Char. | Write Only | 2D | 5A | 2D |
| 3 | R3 | Horiz. Sync Width | Char. | Write Only | 0A | 0A | 0A |
| 4 | R4 | Vertical Total | Char. Row | Write Only | 1F | 1F | 7F |
| 5 | R5 | Vertical Total Adjust | Scan Line | Write Only | 06 | 06 | 06 |
| 6 | R6 | Vertical Displayed | Char. Row | Write Only | 19 | 19 | 64 |
| 7 | R7 | Vert. Sync Position | Char. Row | Write Only | 1C | 1C | 70 |
| 8 | R8 | Interface Mode | — | Write Only | 02 | 02 | 02 |
| 9 | R9 | Max Scan Line Addr. | Scan Line | Write Only | 07 | 07 | 01 |
| A | R10 | Cursor Start | Scan Line | Write Only | 06 | 06 | 06 |
| B | R11 | Cursor End | Scan Line | Write Only | 07 | 07 | 07 |
| C | R12 | Start Addr. (H) | — | Write Only | 00 | 00 | 00 |
| D | R13 | Start Addr. (L) | — | Write Only | 00 | 00 | 00 |
| E | R14 | Cursor Addr. (H) | — | Read/Write | XX | XX | XX |
| F | R15 | Cursor Addr. (L) | — | Read/Write | XX | XX | XX |
| 10 | R16 | Light Pen (H) | — | Read Only | XX | XX | XX |
| 11 | R17 | Light Pen (L) | — | Read Only | XX | XX | XX |

Note: All register values are given in hexadecimal.

Programming the Mode Control and Status Register

The following I/O devices are defined on the Color/Graphics Adapter.

| HEX ADDR. | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | FUNCTION OF REGISTER |
|-----------|----|----|----|----|----|----|----|----|----|----|-------------------------|
| X'308' | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 00 REG (MODE CONTROL) |
| X'309' | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 00 REG (COLOR SELECT) |
| X'30A' | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 01 REG (STATUS) |
| X'30B' | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | CLEAR LIGHT PEN LATCH |
| X'30C' | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | PRE SET LIGHT PEN LATCH |
| X'30D' | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 2 | Z | 0 | 6845 REGISTERS |
| X'30E' | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 2 | Z | 1 | 6845 REGISTERS |
| X'30F' | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 2 | Z | 0 | 6845 REGISTERS |
| X'301' | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 2 | Z | 1 | 6845 REGISTERS |

Z = don't care condition

Color Select Register

This is a 6 bit output only, register, it cannot be read, its address is X'3D9' and can be written using the 8088 I/O OUT command.

The following is a description of the Register functions.

| Bit 0 | B (BLUE) Border Color Select ALPHA/BACKGROUND |
|-------|---|
| Bit 1 | G (GREEN) Border Color Select ALPHA/BACKGROUND |
| Bit 2 | R (RED) Border Color Select ALPHA/BACKGROUND |
| Bit 3 | I Intensifies Border Color Select ALPHA/BACKGROUND IN 320 x 200 |
| Bit 4 | Select Alt Back Color Set For Alpha Color Modes |
| Bit 5 | 320 x 200 Color Set Select |
| Bit 6 | Not Used |
| Bit 7 | Not Used |

Bits 0, 1, 2, 3. Select the screens border color in 40x25 alpha mode. In graphics mode (medium resolution) 320x200 color, the screen background color (C0-C1) is selected by these bit settings.

Bit 4. This bit when set will select on alternate, intensified, set of background colors in the alpha mode.

Bit 5 is only used in the medium resolution color mode (320x200). It is used to select the active set of screen colors for the display.

When bit 5 is set to a "1" colors are determined as follows:

| The C1 | C0 | Set selected are: |
|--------|----|--|
| 0 | 0 | Background as defined by Bit 0-3 of Port '3D9' |
| 0 | 1 | Cyan |
| 1 | 0 | Magenta |
| 1 | 1 | White |

When bit 5 is set to a "0" colors are determined as follows:

| The C0 | C1 | Set selected are: |
|--------|----|--|
| 0 | 0 | Background as defined by Bit 0-3 of Port '3D9' |
| 0 | 1 | Green |
| 1 | 0 | Red |
| 1 | 0 | Yellow |

Mode Select Register

This is a 6 bit output only register, it cannot be read. Its address is X'3D8'. It can be written using the 8088 I/O OUT command.

The following is a description of the registers functions.

| | |
|-------|--|
| Bit 0 | 80x25 mode |
| Bit 1 | Graphic Select |
| Bit 2 | B & W Select |
| Bit 3 | Enable Video Signal |
| Bit 4 | High Res 640x200 B & W Mode |
| Bit 5 | Change BACKGROUND INTENSITY to Blink Bit |
| Bit 6 | Not Used |
| Bit 7 | Not Used |

Bit 0 Selects between 40x25 and 80x25 alpha mode, a "1" sets it to 80x25 mode.

Bit 1 Selects between ALPHA mode and 320x200 graphics mode, a "1" select 320x200 graphics mode.

Bit 2 Selects color or B & W mode, a "1" selects B & W.

Bit 3 Enables the video signal at certain times when modes are being changed. The video signal should be disabled when changing modes. A "1" enables the video signal.

Bit 4 When on, this bit selects the 640x200 B & W graphics mode. One color of 8 can be selected on direct drive sets in this mode by using register 3D9.

Bit 5

When on, this bit will change the character background intensity to the blinking attribute function for ALPHA modes. When the high order attribute bit is not selected, 16 background colors (or intensified colors) are available. For normal operation, this bit should be set to "1" to allow the blinking function.

Mode Register Summary

Bits

| 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | z |
| 0 | 1 | 0 | 1 | 0 | z |
| 0 | 1 | 1 | 1 | 1 | z |

40 x 25 ALPHA B & W

40 x 25 ALPHA COLOR

80 x 25 ALPHA B & W

80 x 25 ALPHA COLOR

320 x 200 B & W GRAPHICS

320 x 200 COLOR GRAPHICS

640 x 200 B & W GRAPHICS

ENABLE BLINK ATTRIBUTE

640 x 200 B & W

ENABLE VIDEO

SELECT B & W MODE

SELECT 320 x 200 GRAPHICS

80 x 25 ALPHA SELECT

z = don't care condition

* THE LOW RESOLUTION 160 x 100 MODE REQUIRES SPECIAL PROGRAMMING AND IS SET UP AS ALPHA MODE 40 x 25

Status Register

The status register is a 4 bit read only register. Its address is X'3DA'. It can be read using the 8088 I/O IN instruction.

The following is a description of the register functions.

- Bit 0 Display Enable
- Bit 1 Light Pen Trigger Set
- Bit 2 Light Pen SW Made
- Bit 3 Alpha Dots
- Bit 4 Not Used
- Bit 5 Not Used
- Bit 6 Not Used
- Bit 7 Not Used
- Bit 0 This input bit, when active, indicates that a regen buffer memory access can be made without interfering with the Display.
- Bit 1 This bit, when active, indicates that a positive going edge from the light pen input has set the light pen trigger. This trigger is reset on power on and may also be cleared by doing an I/O OUT command to address X'3DB'. No specific data setting is required, the action is address activated.
- Bit 2 The light pen switch status is reflected in this status bit. The switch is not latched or debounced. A "0" indicates the switch is on.
- Bit 3 The ALPHA video output signal is readable in this status bit. Its purpose is to verify that video information is being generated for RAS purposes.

Sequence of Events

1. Determine mode of operation
2. Reset Video Enable bit
3. Program 6845 to select mode
4. Program mode/color select registers

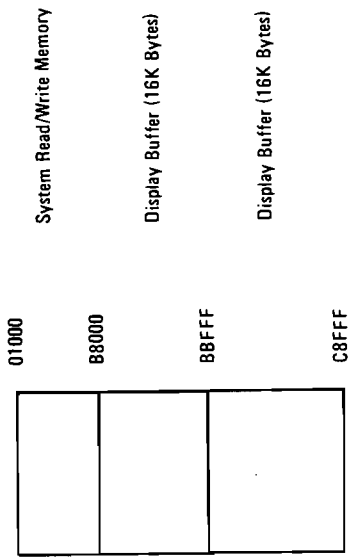
Memory Requirements

The memory used by this adapter is self-contained. It consists of 16K bytes of memory without parity. This memory is used as both a display buffer for alphanumeric data and as a bit map for graphics data. The Regen Buffers address starts at X'B8000'.

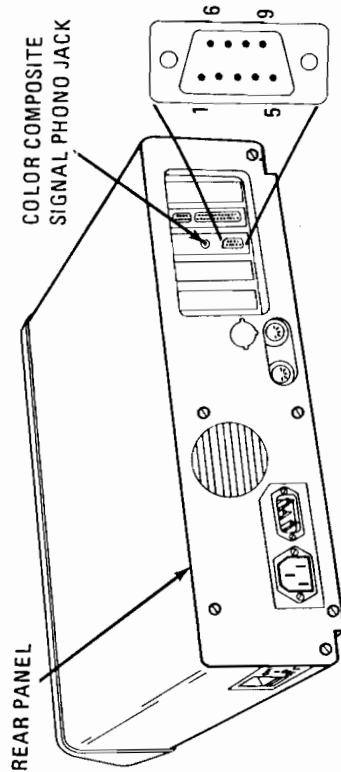
Interrupt Level (Vertical Retrace)

Level 2

I/O Address and Bit Map
Read/Write Memory Address Space

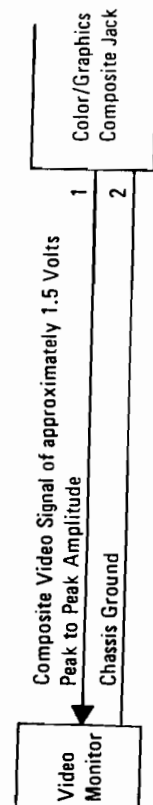
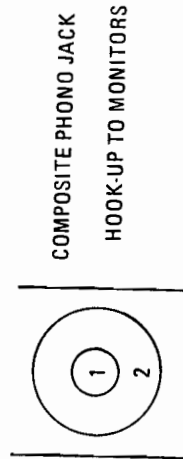
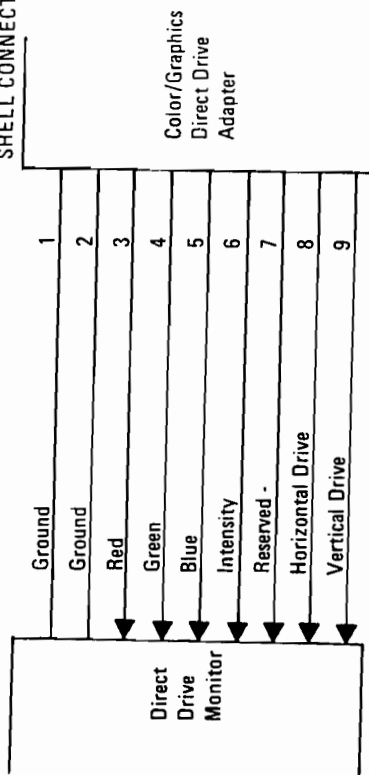


Color/Graphics Monitor Adapter Direct Drive and Composite Interface Pin Assignment

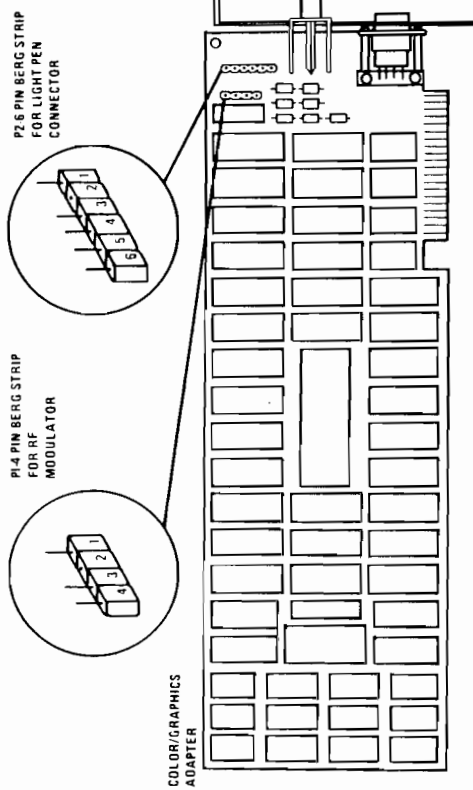


AT STANDARD TTL LEVELS

COLOR DIRECT DRIVE 9 PIN "D" SHELL CONNECTOR

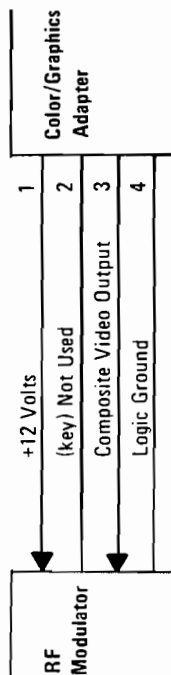


Color/Graphics Monitor Adapter Auxiliary Video Connectors

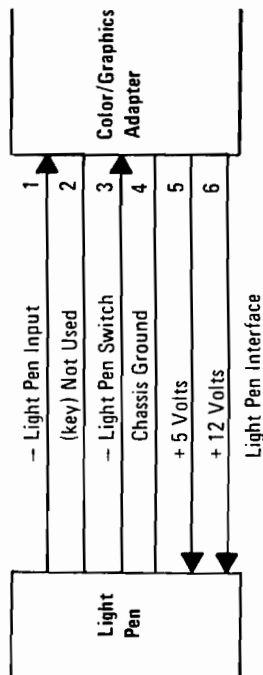


PI-4 PIN BERG STRIP FOR RF MODULATOR

P2 6 PIN BERG STRIP FOR LIGHT PEN CONNECTOR



RF Modulator Interface



Light Pen Interface

Parallel Printer Adapter

The Printer Adapter is specifically designed to attach printers with a parallel port interface, but it can be used as a general input/output port for any device or application which matches its input/output capabilities. It has 12 TTL buffer output points which are latched and can be written and read under program control using the processor IN or OUT instructions. The adapter also has five steady state input points that may be read using the processor's IN instructions.

In addition, one input can also be used to create a processor interrupt. This interrupt can be enabled and disabled under program control. Reset from the power-on circuit is also "ORed" with a program output point allowing a device to receive a power-on reset when the processor is reset.

This function is packaged on an adapter which fits into any of the five System Expansion slots on the System Board. The input/output signals are made available at the back of the adapter via a right angle PCB mounted 25 PIN "D" type connector. This connector protrudes through the rear panel of the System Unit where a cable and shield may be attached.

When this adapter is used to attach a printer, data, or printer, commands are loaded into an 8-bit latched output port, and the strobe line is activated writing data to the printer. The program then may read the input ports for printer status indicating when the next character can be written or it may use the interrupt line to indicate "not busy" to the software.

The output ports may also be read at the card's interface for diagnostic loop functions. This allows fault isolation determination between the adapter and the attaching device.

This same function is also part of the combination IBM Monochrome Display and Printer Adapter. A block diagram of the printer adapter is on the following page.

Programming Considerations

The Printer Adapter responds to 5 I/O instructions — 2 output and 3 input. The output instructions transfer data into 2 latches whose outputs are presented on pins of a 25 Pin "D" shell connector.

Two of the three input instructions allow the CPU to read back the contents of the two latches. The third allows the CPU to read the real time status of a group of pins on the connector. A description of each instruction follows.

| IBM Monochrome Display & Printer Adapter | | | | Parallel Printer Adapter | | | |
|--|-------|-------|-------|--------------------------|-------|-------|-------|
| Output to address 3BCH | | | | Output to address 378H | | | |
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| Pin 9 | Pin 8 | Pin 7 | Pin 6 | Pin 5 | Pin 4 | Pin 3 | Pin 2 |

This instruction captures data from the data bus and is present on the respective pins. These pins are each capable of sourcing 2.6 ma and sinking 24 ma.

It is essential that the external device not try to pull these lines to ground.

| IBM Monochrome Display & Printer Adapter | Parallel Printer Adapter |
|--|--------------------------|
| Output to address 38EH | Output to address 37AH |
| Bit 4 | Bit 3 |
| IRQ | Bit 2 |
| Enable | Bit 1 |
| | Bit 0 |
| | Pin 14 |
| | Pin 16 |
| | Pin 17 |
| | Pin 1 |

This instruction causes this latch to capture the five least significant bits of data bus. The four least significant bits present their outputs, or inverted versions of their outputs to the respective pins shown above. If bit 4 is written 1, the card will interrupt the CPU on the condition that Pin 10 transitions high to low.

These pins are driven by open collector drivers pulled to +5V through 4.7K OHM resistors. They can each sink approximately 7 ma and maintain 0.8 volts down level.

Note: For pin references, see Parallel Interface Connector Specifications, page 2-79.

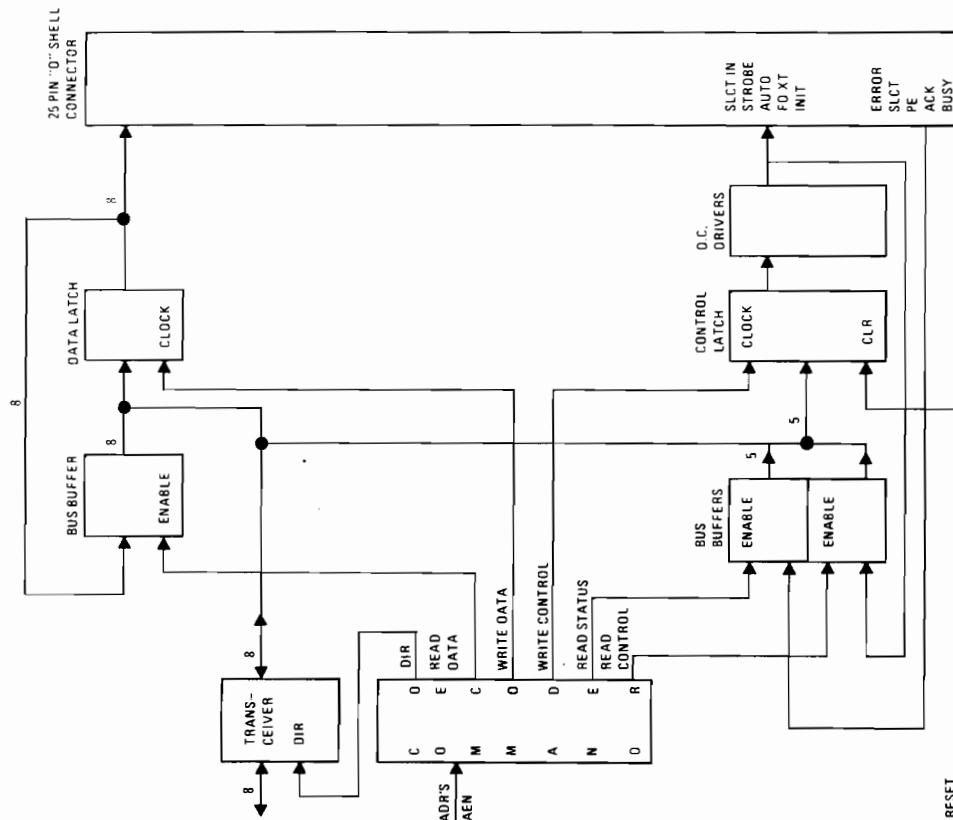


Figure 16. PARALLEL PRINTER ADAPTER BLOCK DIAGRAM

| IBM Monochrome Display & Printer Adapter | |
|--|---|
| Input from address x' 3BC' | Parallel Printer Adapter Input from address 378H |

This command presents the CPU with data present on the pins associated with the out to x' 3BC'. This should normally reflect the exact value that was last written to x' 3BC'. If an external device should be driving data on these pins (in violation of usage ground rules) at the time of an input, this data will be 'or'ed with the latch contents.

| IBM Monochrome Display & Printer Adapter | |
|--|---|
| Input from address 38DH | Parallel Printer Adapter Input from address 379H |

This command presents real time status to the CPU from the pins as follows.

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---------|--------|--------|--------|--------|-------|-------|-------|
| Pin 11* | Pin 10 | Pin 12 | Pin 13 | Pin 15 | — | — | — |

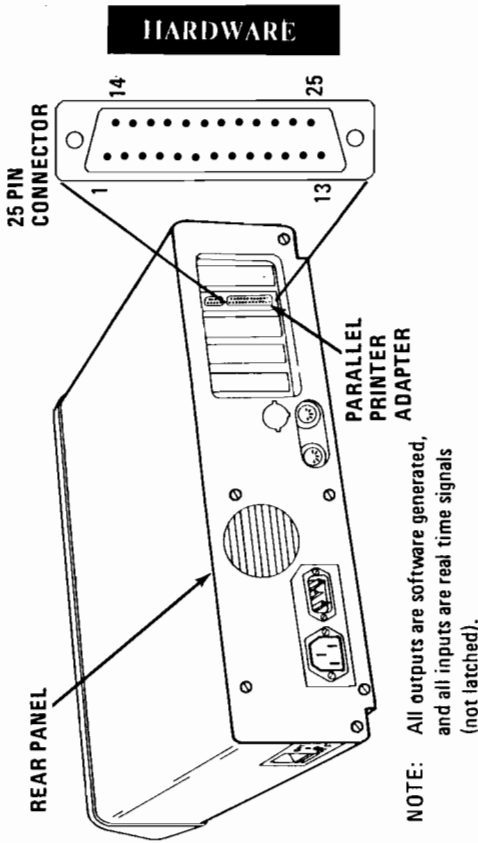
| IBM Monochrome Display & Printer Adapter | |
|--|---|
| Input from address 38EH | Parallel Printer Adapter Input from address 37AH |

This instruction causes the data present on pins 1, 14, 16, 17 and IRQ bit to be read by the CPU. In the absence of external drive applied to these pins, data read by the CPU will exactly match data last written to x' 3BE' in the same bit positions. Note that data bits 0-2 are not included. If external drivers are dotted to these pins, that data will be 'or'ed with data applied to the pins by the x' 3BE' latch.

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|------------|--------|--------|--------|-------|
| | | | IRQ Enable | Pin 17 | Pin 16 | Pin 14 | Pin 1 |
| | | | Por=0 | Por=1 | Por=0 | Por=1 | Por=1 |

These pins assume the states shown after a reset from the CPU.
Note: For pin references see Parallel Printer Adapter Interface Connector Specifications page 2-79.

Parallel Printer Adapter Interface Connector Specifications



NOTE: All outputs are software generated, and all inputs are real time signals (not latched).

AT STANDARD TTL LEVELS

| Signal Name | Adapter Pin No. |
|-------------------------|-----------------|
| — Strobe | 1 |
| + Data Bit 0 | 2 |
| + Data Bit 1 | 3 |
| + Data Bit 2 | 4 |
| + Data Bit 3 | 5 |
| + Data Bit 4 | 6 |
| + Data Bit 5 | 7 |
| + Data Bit 6 | 8 |
| + Data Bit 7 | 9 |
| — Acknowledge | 10 |
| + Busy | 11 |
| + P. End (out of Paper) | 12 |
| + Select | 13 |
| — Auto Feed | 14 |
| — Error | 15 |
| — Initialize Printer | 16 |
| — Select Input | 17 |
| Ground | 18 - 25 |

IBM 80 CPS Graphics Printer

The printer is a self powered, standalone, table top unit. It attaches to the system unit via a parallel signal cable 6 feet (1.8 meters) in length. It obtains AC power from a standard wall outlet and is available in three different versions (120, 220, or 240 Vac). The printer is an 80 Character Per Second (CPS) bidirectional wire matrix device. It prints characters in a 9x9 dot matrix with a 9 wire head. It can print in a compressed mode of 132 characters per line and in a standard font, 80 characters per line. A large font also prints in 66 characters per line mode.

The printer can print double size characters and double strike characters. It can select from two character sets, an extended upper/lower case character set for international languages, and the U.S. standard upper/lower case 96 character ASCII character set. It can print subscript, superscript, defined graphics characters, underlining, and programmable graphics.

The printer can also accept commands setting the line feed control desired for the application. It attaches to the system unit via the Parallel Printer Adapter or the IBM Monochrome Display and Printer Adapter. The cable is a 25 lead shielded cable with a 25 pin "D" type connector at the system unit end, and a 36 pin connector on the printer end.

Table 7. Printer Specifications

| (1) | Print Method: | Serial Impact dot matrix | | | | | | | | | | | | |
|---------------------|-----------------------------|---|---------------------|-----------------------------|----|----|---|----|------|-----|------|----|----|----|
| (2) | Print Speed: | 80 CPS | | | | | | | | | | | | |
| (3) | Print Direction: | Bidirectional with logical seeking | | | | | | | | | | | | |
| (4) | Number of Pins in Head | 9 | | | | | | | | | | | | |
| (5) | Line Spacing: | 4.23 mm (1/6 inch) or programmable | | | | | | | | | | | | |
| (6) | Printing Characteristics | | | | | | | | | | | | | |
| | Matrix: | 9x9 | | | | | | | | | | | | |
| | Character Set 1: | Full 96-character ASCII with descenders, additional ASCII numbers 160 to 175 are European characters. 176 to 223 are graphic characters. 224 to 239 are selected Greek characters. 240 to 255 contain math and extra symbols. | | | | | | | | | | | | |
| | Character Set 2: | Same as above except: ASCII numbers 3,4,5, 6, and 21 contain symbols. ASCII numbers 128 to 175 contain European Characters. | | | | | | | | | | | | |
| (7) | Graphics: | 20 block characters and programmable graphics. | | | | | | | | | | | | |
| | Printing Sizes | <table><tr><th>Characters per inch</th><th>Maximum characters per line</th></tr><tr><td>10</td><td>80</td></tr><tr><td>5</td><td>40</td></tr><tr><td>16.5</td><td>132</td></tr><tr><td>8.25</td><td>66</td></tr><tr><td>10</td><td>80</td></tr></table> | Characters per inch | Maximum characters per line | 10 | 80 | 5 | 40 | 16.5 | 132 | 8.25 | 66 | 10 | 80 |
| Characters per inch | Maximum characters per line | | | | | | | | | | | | | |
| 10 | 80 | | | | | | | | | | | | | |
| 5 | 40 | | | | | | | | | | | | | |
| 16.5 | 132 | | | | | | | | | | | | | |
| 8.25 | 66 | | | | | | | | | | | | | |
| 10 | 80 | | | | | | | | | | | | | |
| (8) | Media Handling | | | | | | | | | | | | | |
| | Paper Feed: | Adjustable sprocket pin feed | | | | | | | | | | | | |
| | Paper Width Range: | 101.6 mm (4 inches) to 254 mm (10 inches) | | | | | | | | | | | | |
| | Copies: | One original plus two carbon copies, total thickness not to exceed 0.3 mm (0.012 inch). Minimum paper thickness is 0.064 mm or 0.0025 inch. | | | | | | | | | | | | |
| (9) | Paper Path: | Rear | | | | | | | | | | | | |
| | Interfaces Standard: | Parallel 8-bit Data Data and Control lines | | | | | | | | | | | | |
| (10) | Inked Ribbon | | | | | | | | | | | | | |
| | Color: | Black | | | | | | | | | | | | |
| | Type: | Cartridge | | | | | | | | | | | | |
| | Life Expectancy: | 3 million characters | | | | | | | | | | | | |

Table 7. Printer Specifications (continued)

| | | | | |
|------|--------------------------|--------------------------|-------|-------|
| (11) | Environmental Conditions | 5 to 35°C (41 to 95° F) | | |
| | | 10 to 80% non-condensing | | |
| (12) | Power Requirements | Three models available: | | |
| | | 120 | 220 | 240 |
| | Voltage (AC): | 60 | 50/60 | 50/60 |
| | | 1 | .5 | .5 |
| | Frequency (Hz): | 100 | 100 | 100 |
| | | | | |
| (13) | Physical Characteristics | 107 mm (4.2 inches) | | |
| | | 374 mm (14.7 inches) | | |
| | Height: | 305 mm (12.0 inches) | | |
| | | 5.5 kg (12 lbs.) | | |
| | Width: | | | |
| | | | | |
| | Depth: | | | |
| | | | | |
| | Weight: | | | |
| | | | | |

Setting the DIP Switches

There are two DIP switches on the control circuit board. In order to suit the user's specific requirements, desired control modes are selectable by the DIP switches. The functions of the switches and their preset conditions at the time of shipment are as shown in Table 8 (DIP Switch 1) and Table 9 (DIP Switch 2).

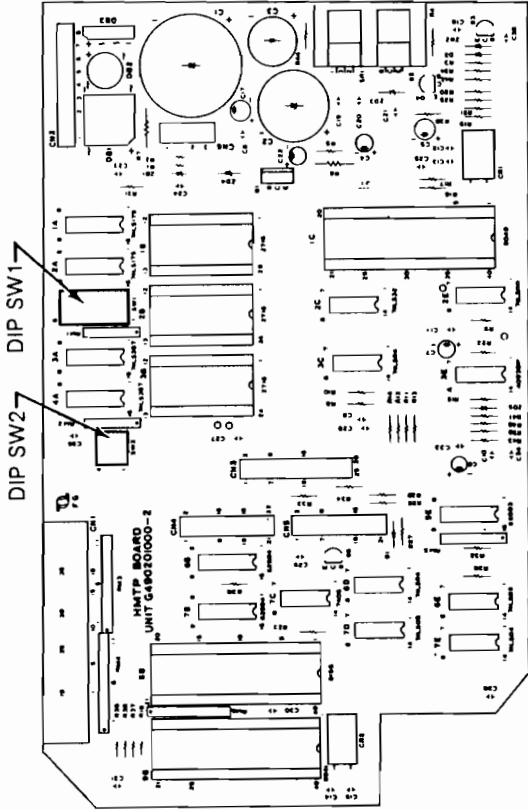


Figure 17. LOCATION OF PRINTER DIP SWITCHES

Table 8. Functions and Conditions of DIP Switch 1

| Switch No. | Function | ON | OFF | Factory-set Condition |
|------------|---------------------|------------------|----------------------|-----------------------|
| 1-1 | Not applicable | — | — | ON |
| 1-2 | CR | Print only | Print & line feed | ON |
| 1-3 | Buffer full | Print only | Print & line feed | OFF |
| 1-4 | Cancel code | Invalid | Valid | OFF |
| 1-5 | Not applicable | — | — | ON |
| 1-6 | Error buzzer | Sound | Does not sound | ON |
| 1-7 | Character generator | Table 2 | Table 1 | ON |
| 1-8 | Select in signal | Fixed internally | Not fixed internally | ON |

Table 9. Functions and Conditions of DIP Switch 2

| Switch No. | Function | ON | OFF | Factory-set Condition |
|------------|------------------------------|------------------|----------------------|-----------------------|
| 2-1 | Form length | 12" 304.8 mm | 11" 279.4 mm | OFF |
| 2-2 | Line spacing | 1/8" 3.175 mm | 1/6" 4.23 mm | OFF |
| 2-3 | Auto Feed XT signal | Fixed internally | Not fixed internally | OFF |
| 2-4 | 1 inch skip over perforation | Valid | Not valid | OFF |

Parallel Interface Description

- (1) Specifications
 - (a) Data transfer rate: 1000 CPS (max.)
 - (b) Synchronization: By externally supplied STROBE pulses.
 - (c) Handshaking: ACKNLG or BUSY signals.
 - (d) Logic level: Input data and all interface control signals are compatible with the TTL level.
- (2) Connector
 - Plug: 57-30360 (AMPHENOL)
- (3)
 - Connector pin assignment and descriptions of signals.
 - Connector pin assignment and descriptions of respective interface signals are provided in Table (10) page 2-86.

Table 10. Connector Pin Assignment and Description of Interface Signals

| Signal Pin No. | Return Pin No. | Signal | Direction | Description |
|----------------|----------------|--------|-----------|--|
| 1 | 19 | STROBE | In | STROBE pulse to read data in. Pulse width must be more than 0.5 μ s at receiving terminal. The signal level is normally "HIGH"; read-in of data is performed at the "LOW" level of this signal. |
| 2 | 20 | DATA 1 | In | These signals represent information of the 1st to 8th bits of parallel data respectively. Each signal is at "HIGH" level when data is logical "1" and "LOW" when logical "0." |
| 3 | 21 | DATA 2 | In | |
| 4 | 22 | DATA 3 | In | |
| 5 | 23 | DATA 4 | In | |
| 6 | 24 | DATA 5 | In | |
| 7 | 25 | DATA 6 | In | |
| 8 | 26 | DATA 7 | In | |
| 9 | 27 | DATA 8 | In | |
| 10 | 28 | ACKNLG | Out | Approx. 5 μ s pulse. "LOW" indicates that data has been received and that the printer is ready to accept other data. |
| 11 | 29 | BUSY | Out | A "HIGH" signal indicates that the printer cannot receive data. The signal becomes "High" in the following cases: 1. During data entry 2. During printing operation 3. In OFF-LINE state 4. During printer error status. |

Table 10. Connector Pin Assignment and Description of Interface Signals (continued)

| Signal Pin No. | Return Pin No. | Signal | Direction | Description |
|----------------|----------------|--------------|-----------|--|
| 12 | 30 | PE | Out | A "HIGH" signal indicates that the printer is out of paper. |
| 13 | — | SLCT | Out | This signal indicates that the printer is in the selected state. |
| 14 | — | AUTO FEED XT | In | With this signal being at "LOW" level, the paper is automatically fed one line after printing. (The signal level can be fixed to "LOW" with DIP SW pin 2-3 provided on the control circuit board.) |
| 15 | — | NC | | Not used. |
| 16 | — | OV | | Logic GND level. |
| 17 | — | CHASSIS-GND | — | Printer chassis GND. In the printer, the chassis GND and the logic GND are isolated from each other. |
| 18 | — | NC | — | Not used. |
| 19-30 | — | GND | — | TWISTED-PAIR RETURN signal GND level. |
| 31 | — | INIT | In | When the level of this signal becomes "LOW" the printer controller is reset to its initial state and the print buffer is cleared. This signal is normally at "HIGH" level, and its pulse width must be more than 50 μ s at the receiving terminal. |

Table 10. Connector Pin Assignment and Description of Interface Signals (continued)

| Signal Pin No. | Return Pin No. | Signal | Direction | Description |
|----------------|----------------|---------|-----------|--|
| 32 | | ERROR | Out | The level of this signal becomes "LOW" when the printer is in— 1. PAPER END state 2. OFF-LINE state 3. Error state |
| 33 | — | GND | — | Same as with Pin No. 19 to 30. |
| 34 | — | NC | — | Not used. |
| 35 | | | | Pulled up to +5V through 4.7K Ω resistance. |
| 36 | — | SLCT IN | In | Data entry to the printer is possible only when the level of this signal is "LOW". (Internal fixing can be carried out with DiP SW 1-8. The condition at the time of shipment is set "LOW" for this signal.) |

NOTES 1: "Direction" refers to the direction of signal flow as viewed from the printer.

2: "Return" denotes "TWISTED PAIR RETURN" and is to be connected at signal ground level.

As to the wiring for the interface, be sure to use a twisted-pair cable for each signal and never fail to complete connection on the Return side. To prevent noise effectively, these cables should be shielded and connected to the chassis of the System Unit and the printer, respectively.

3: All interface conditions are based on TTL level. Both the rise and fall times of each signal must be less than 0.2 μ s.

4: Data transfer must not be carried out by ignoring the ACKNLG or BUSY signal. (Data transfer to this printer can be carried out only after confirming the ACKNLG signal or when the level of the BUSY signal is "LOW".)

- (4) Data transfer sequence
Figure 17 shows the sequence for data transmission.

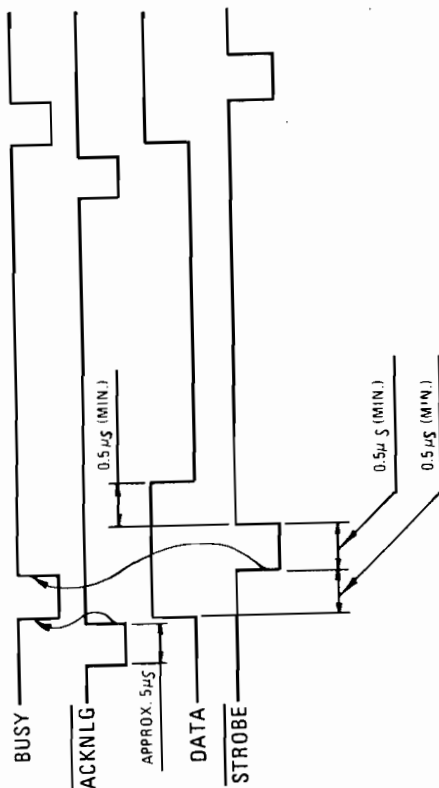


Figure 18. PARALLEL INTERFACE TIMING DIAGRAM

Below are the allowed combinations of print modes that can be selected.

The IBM 80 CPS Graphics Printer can select any of the combinations listed below and may change print modes at any place within a line.

Note: Modes can be selected and combined if they are in the same vertical column.

[illegible]

Printer Control Codes

On the following pages you will find complete codes for printer control, characters, and graphics. You may want to keep them handy for future reference. The printer codes are listed in ASCII decimal order (e.g. NUL is zero, BEL is 7, etc.). Examples given in the Printer Function descriptions are written in BASIC. The "Format" description is given when more information is needed for programming considerations. ASCII decimal values for the printer control codes can be found in the Printer Character Sets which follow these control codes.

| Printer Code | Printer Function |
|--------------|--|
| NUL | <p>Null</p> <p>Used with ESC B and ESC D as a list terminator. NUL is also used with other Printer Control Codes to select options (e.g. ESC S).</p> <p>Example: LPRINT CHR\$(0);</p> |
| BEL | <p>Bell</p> <p>Sounds the printer buzzer for 1 second.</p> <p>Example: LPRINT CHR\$(7);</p> |
| HT | <p>Horizontal Tab</p> <p>Tabs to the horizontal tab stop. When the printer is powered on or reset, tab stops are set every eight columns. Tab stops can be changed with the ESC D command.</p> <p>Example: LPRINT CHR\$(9)</p> |
| LF | <p>Line Feed</p> <p>Prints data remaining in the printer buffer and spaces the paper up one line. Line spacing is 1/6 inch (4.23 mm) unless reset by ESC A, ESC 0, ESC 1, ESC 2, or ESC 3.</p> <p>Example: LPRINT CHR\$(10);</p> |

| Printer Code | Printer Function | Printer Code | Printer Function |
|--------------|---|--------------|---|
| VT | Vertical Tab The VT code is treated as a LF. Example: LPRINT CHR\$(11); | SI | Shift In (Compressed) Changes the printer to the Compressed Character print mode. Example: LPRINT CHR\$(15); |
| FF | Form Feed Advances the paper to the top of the next page. The location of the paper, when the printer is powered on or reset, determines the top of the page. The next top of page is 11 inches (279.4 mm) from that position. ESC C can be used to change the page length. Example: LPRINT CHR\$(12); | DC2 | Device Control 2 (Compressed Off) Stops printing in the Compressed Character print mode. Example: LPRINT CHR\$(18); |
| CR | Carriage Return Ends the line the printer is on and prints the data remaining in the printer buffer. (No line feed operation is initiated by the printer.) Example: LPRINT CHR\$(13); | DC4 | Device Control 4 (Double Width Off) Stops printing in the Double Width print mode started by the SO command. Example: LPRINT CHR\$(20); |
| | IBM Personal Computer Basic adds a Line Feed command to every Carriage Return unless you select character set 1, add 128 to the CR, and return to character set 2. (See ESC 6 and ESC 7.) Example: LPRINT "0000";CHR\$(27);CHR\$(55);CHR\$(13+128);CHR\$(27);CHR\$(54);"////" | CAN | Cancel Clears the printer buffer without printing data. Control codes, except SO, remain in effect. Example: LPRINT CHR\$(24); |
| | | ESC | Escape Lets the printer know that the next data sent is a printer command. (See the following list of commands.) Example: LPRINT CHR\$(27); |
| SO | Shift Out (Double Width) Changes to the Double Width print mode. A Carriage Return, Line Feed, DC4, or CAN cancels Double Width print mode initiated with SO. See ESC W for continuous Double Width printing. Example: LPRINT CHR\$(14); | ESC - | Escape Minus (Underline) Format: ESC -n; ESC - followed by a 1, prints all of the following data with an underline. ESC - followed by a 0 (zero), cancels the Underline print mode. Example: LPRINT CHR\$(27);CHR(45);CHR\$(1); |

| Printer Code | Printer Function | Printer Code | Printer Function |
|--------------|--|--------------|--|
| ESC 0 | Escape Zero (1/8-inch Line Feeding) Changes paper feeding to 1/8 inch (3.175 mm). Example: LPRINT CHR\$(27);CHR\$(48); | ESC 8 | Escape Eight (Ignore Paper End) Allows the printer to print to the end of the paper. The printer ignores the paper end switch. This command must be sent prior to an actual paper end. Example: LPRINT CHR\$(27);CHR\$(56); |
| ESC 1 | Escape 1 (7/72-inch Line Feeding) Changes paper feed to 7/72 inch (2.47 mm). Example: LPRINT CHR\$(27);CHR\$(49); | ESC 9 | Escape Nine (Cancel Ignore Paper End) Cancels the Ignore Paper End command. ESC 9 is selected when the printer is powered on or reset. Example: LPRINT CHR\$(27);CHR\$(57); |
| ESC 2 | Escape Two (Starts Variable Line Feeding) ESC 2 is an execution command for ESC A. If no ESC A command has been given, line feeding returns to 1/6 inch (4.23 mm). Example: LPRINT CHR\$(27);CHR\$(50); | ESC < | Escape Less Than (Home Head) The print head will return to the left margin to print the line following ESC <. This will occur for one line only. Example: LPRINT CHR\$(27);CHR\$(60); |
| ESC 3 | Escape Three (Variable Line Feeding) Format: ESC 3;n; (Graphics Printer only) Changes the paper feeding to n/216-inch (1/216 inch is 0.1176 mm). A value from 1 to 255 must be assigned to n. The example below sets the paper feeding to 54/216 (1/4) inch (6.35 mm). Example: LPRINT CHR\$(27);CHR\$(51);CHR\$(54); | ESC A | Escape A (Sets the variable line feeding) Format: ESC A;n; Escape A sets the line feed feeding to n/72 inch (1/72 inch is 0.3528 mm). The example below tells the printer to set line feeding to 24/72 inch (8.47 mm). ESC 2 must be sent to the printer before the line feeding will change, e.g. ESC A;24 (text) ESC 2 (text). The text following ESC A;24 will space at the previously set line-feed increments. The text following ESC 2 will be printed with new line feed increments of 24/72 inch (8.47 mm). Any increment between 1/72 and 85/72 may be used. The default for this command at power on or reset is 6/72 inch (4.23 mm). Example: LPRINT CHR\$(27);CHR\$(65);CHR\$(24);CHR\$(27);CHR\$(50); |
| ESC 6 | Escape Six (Select Character Set 2) Selects character set 2. (See Printer Character Set 2.) Character Set 2 is selected when the printer is powered on or reset. Example: LPRINT CHR\$(27);CHR\$(54); | | |
| ESC 7 | Escape Seven (Select Character Set 1) Selects character set 1. (See Printer Character Set 1.) Example: LPRINT CHR\$(27);CHR\$(55); | | |

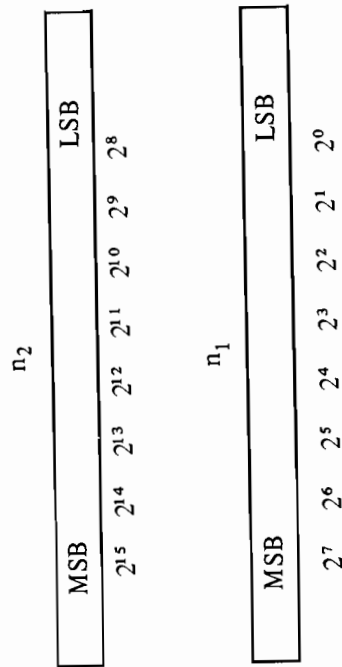
| Printer Code | Printer Function |
|--------------|--|
| ESC C | <p>Escape C (Set lines per page) Format: ESC C;n; Sets the page length in number of lines. The ESC C command must have a value (from 1 to 127) following it to specify the number of lines desired on the page. The example below sets the page length to 55 lines. The printer defaults to 66 lines per page when powered on or reset. Example: LPRINT CHR\$(27);CHR\$(67);CHR\$(55);</p> <p>Escape C (Set inches per page) Format: ESC C;n;m; Sets the length of the page in inches (one inch is 25.4 mm). This command requires a value of 0 (zero) for n and a value between 1 and 22 for m. Example: LPRINT CHR\$(27);CHR\$(67);CHR\$(0);CHR\$(12);</p> <p>Escape D (Set Horizontal Tab Stops) Format: ESC D;n₁;n₂...n_k;NUL; Sets the horizontal tab stop positions. The example below shows the horizontal tab stop positions set at printer column positions of 10, 20 and 40. They are followed by CHR\$(0), the NUL code. They must be in ascending numerical order as shown. Tab stops can be set between 1 and 80 in normal print mode. When in the compressed print mode, tab stops can be set up to 132. Double Width characters take up two column positions. The maximum number of tabs that can be set is 28. The HT [CHR\$(9)] is used to execute a tab operation. Example: LPRINT CHR\$(27);CHR\$(68);CHR\$(10);CHR\$(20);CHR\$(40);CHR\$(0);</p> |
| ESC D | |

| Printer Code | Printer Function |
|--------------|---|
| ESC E | <p>Escape E (Emphasized) Changes the printer to the Emphasized print mode. The printer speed is reduced to half during the Emphasized print mode. Example: LPRINT CHR\$(27);CHR\$(69);</p> <p>Escape F (Emphasized Off) Stops printing in the Emphasized print mode. Example: LPRINT CHR\$(27);CHR\$(70);</p> <p>Escape G (Double Strike) Changes the printer to the Double Strike print mode. The paper is spaced 1/216 inch (0.1176 mm) before the second pass of the print head. Example: LPRINT CHR\$(27);CHR\$(71);</p> <p>Escape H (Double Strike Off) Stops printing in the Double Strike mode. Example: LPRINT CHR\$(27);CHR\$(72);</p> <p>Escape J (Feed Paper n/216 inch) Format: ESC J;n; When ESC J is sent to the printer, the paper will be fed n/216 of an inch. (1/216 inch is 0.1176 mm.) The value of "n" must be between 1 and 255. The example below produces a paper feed of 50/216 inch (5.88 mm). ESC J is canceled after the paper feed takes place. Example: LPRINT CHR\$(27);CHR\$(74);CHR\$(50);</p> |
| ESC F | |
| ESC G | |
| ESC H | |
| ESC J | |

Printer Code
ESC K

Printer Function

Escape K (480 Bit Image Graphics Mode)
Format: ESC K $n_1 n_2 v_1 v_2 \dots v_k$;
Changes from the Text mode to the Bit Image Graphics mode. n_1 and n_2 are numbers, each consisting of 1 byte, which specify the total number of Bit Image Data bytes to be transferred. v_1 through v_k are the bytes of Bit Image Data whose total number (k) cannot exceed 480 and must be equal to $n_1 + 256n_2$. At every horizontal position each byte can print up to 8 vertical dots. Bit Image Data may be mixed with Text data on the same line.
Note: Assign values to n_1 and n_2 as follows: n_1 represents values from 0 - 255. n_2 represents values from 0 - 1 x 256.
MSB is Most Significant Bit and LSB is Least Significant Bit.



Data sent to the printer.

| Text (20 characters) | ESC K n=360 | Bit-image data | Next data |
|----------------------|-------------|----------------|-----------|
|----------------------|-------------|----------------|-----------|

20 characters in text mode correspond to 120 bit-image positions (20 x 6 = 120). The printable portion left in Bit-image mode is 360 dot positions (480 - 120 = 360).
Data sent to the printer.

| | | n_1, n_2 | | | | | | n_1, n_2 | |
|-----------|-------|----------------|-------|----------------|-----------|-------|----------------|------------|----------------|
| Data A | ESC K | n_1 | n_2 | Data B | Data C | ESC K | n_1 | n_2 | Data D |
| Text data | | Length of data | | Bit-image data | Text data | | Length of data | | Bit-image data |

480 bit-image dot positions

HARDWARE

Example:

```

1  'OPEN PRINTER IN RANDOM MODE WITH LENGTH OF
   255
2  OPEN "LPT1:" AS #1
3  WIDTH "LPT1:" 255
4  PRINT #1,CHR$(13);CHR$(10);
5  SLASH$=CHR$(1)+CHR$(2)+CHR$(4)+CHR$(8)
6  SLASH$=SLASH$+CHR$(16)+CHR$(32)+CHR$(64)
   +CHR$(128)+CHR$(0)
7  GAP$=CHR$(0)+CHR$(0)+CHR$(0)
8  NDOTS=480
9  'ESC K N1 N2
10 PRINT #1,CHR$(27);"K";CHR$(NDOTS MOD 256);
   CHR$(INT(NDOTS/256));
11 'SEND NDOTS NUMBER OF BIT IMAGE BYTES
12 FOR I=1 TO NDOTS/12 'NUMBER OF SLASHES TO
   PRINT USING GRAPHICS
13 PRINT #1,SLASH$;GAP$;
14 NEXT I
15 CLOSE
16 END

```

This example will give you a row of slashes printed in the 480 Bit Image mode.

| Printer Code | Printer Function |
|--------------|--|
| ESC L | <p>Escape L (960 Bit Image Graphics Mode) Format: ESC L;n_1;n_2;v_1;v_2;$...v_k$; Changes from the Text mode to the Bit Image Graphics mode. The input is similar to ESC K. The 960 Bit Image mode prints at half the speed of the 480 Bit Image mode, but can produce a denser graphic image. The number of bytes of Bit Image Data (k) is $n_1 + 256n_2$ but cannot exceed 960. n_1 is in the range of 0 to 255.</p> |
| ESC N | <p>Escape N (Set Skip Perforation) Format: ESC N;n; Sets the Skip Perforation function between pages. The number following ESC N sets the value for the number of lines of Skip Perforation. This value can be from 1 to 127. The example shows a 12-line skip perforation. With a page length of 66 lines this will print 54 lines and feed the paper 12 lines. ESC N is reset anytime the page length (ESC C) is changed. Example: LPRINT CHR\$(27);CHR\$(78);CHR\$(12);</p> |
| ESC O | <p>Escape O (Cancel Skip Perforation) Cancels the Skip Perforation function. Example: LPRINT CHR\$(27);CHR\$(79);</p> |
| ESC S | <p>Escape S (Subscript/Superscript) Format: ESC S;n; Changes the printer to the Subscript print mode when ESC S is followed by a 1 as in the example below. When ESC S is followed by a 0 (zero), the printer will print in the Superscript print mode. Example: LPRINT CHR\$(27);CHR\$(83);CHR\$(1);</p> |
| ESC T | <p>Escape T (Subscript/Superscript Off) The printer stops printing in the Subscript or Superscript print mode. Example: LPRINT CHR\$(27);CHR\$(84);</p> |
| ESC U | <p>Escape U (Unidirectional Printing) Format: ESC U;n; The printer will print only from left to right following the input of ESC U;1. When ESC U is followed by a 0 (zero), the left to right printing operation is canceled. The Unidirectional print mode (ESC U) assures a more accurate printing start position for better print quality. Example: LPRINT CHR\$(27);CHR\$(85);CHR\$(1);</p> |
| ESC W | <p>Escape W (Double Width) Format: ESC W;n; Changes to the Double Width mode when ESC W is followed by a 1. This mode is not canceled by a line feed operation or by the DC4 code. It must be canceled with ESC W followed by a 0 (zero). Example: LPRINT CHR\$(27);CHR\$(87);CHR\$(1);</p> |
| ESC Y | <p>Escape Y (960 Bit Image Graphics Mode) Format: ESC Y n_1;n_2;v_1;v_2;$...v_k$; Changes from the Text mode to the 960 Bit Image Graphics mode. The printer prints at normal speed during this operation and cannot print dots on consecutive dot positions. The input of data is similar to ESC L.</p> |
| ESC Z | <p>Escape Z (1920 Bit Image Graphics Mode) Format: ESC Z;n_1;n_2;v_1;v_2;$...v_k$; Changes from the Text mode to the 1920 Bit Image Graphics mode. The input is similar to the other Bit Image Graphics modes during this operation. ESC Z prints at normal speed but can only print every third dot position.</p> |

Table 11. Graphics Printer Character Set 1

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| NUL | | | | | | | BEL | | HT |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| LF | VT | FF | CR | SO | SI | | DC2 | | |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| DC4 | | | | CAN | | | ESC | | |
| 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 |
| | | SP | ! | " | # | \$ | % | & | ' |
| 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 |
| (|) | * | + | , | - | . | / | 0 | 1 |
| 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; |
| 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 |
| < | = | > | ? | @ | A | B | C | D | E |
| 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 |
| F | G | H | I | J | K | L | M | N | O |
| 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 |
| P | Q | R | S | T | U | V | W | X | Y |
| 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 |
| Z | [| \ |] | ^ | _ | ` | a | b | c |
| 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 |
| d | e | f | g | h | i | j | k | l | m |
| 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 |
| n | o | p | q | r | s | t | u | v | w |
| 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 | 129 |
| x | y | z | { | | } | ~ | | NUL | |

Table 11. Graphics Printer Character Set 1 (continued)

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 |
| | | | | | BEL | | HT | LF | VT |
| 140 | 141 | 142 | 143 | 144 | 145 | 146 | 147 | 148 | 149 |
| FF | CR | SO | SI | | DC2 | | DC4 | | |
| 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 |
| | | CAN | | | ESC | | | | |
| 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 |
| á | í | ó | ú | ñ | Ñ | ä | ö | ¿ | ┐ |
| 170 | 171 | 172 | 173 | 174 | 175 | 176 | 177 | 178 | 179 |
| ┐ | 1/2 | 1/4 | i | << | >> | | | | |
| 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 |
| ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ |
| 190 | 191 | 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 |
| ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ |
| 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 | 208 | 209 |
| ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ |
| 210 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 | 219 |
| ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ |
| 220 | 221 | 222 | 223 | 224 | 225 | 226 | 227 | 228 | 229 |
| ┐ | ┐ | ┐ | ┐ | α | β | Γ | Π | Σ | σ |
| 230 | 231 | 232 | 233 | 234 | 235 | 236 | 237 | 238 | 239 |
| μ | τ | ϑ | Θ | Ω | δ | ∞ | ∅ | ε | ∩ |
| 240 | 241 | 242 | 243 | 244 | 245 | 246 | 247 | 248 | 249 |
| ≡ | ± | ≈ | ≠ | ┐ | ┐ | ┐ | ÷ | ≈ | ◻ |
| 250 | 251 | 252 | 253 | 254 | 255 | | | | |
| - | √ | ∩ | 2 | ■ | SP | | | | |

Table 12. Graphics Printer Character Set 2

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| NUL | | | ♥ | ♦ | ♣ | ♠ | BEL | | HT |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| LF | VT | FF | CR | SO | SI | | | DC2 | |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| DC4 | § | | | CAN | | | ESC | | |
| 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 |
| | | SP | ! | " | # | \$ | % | & | ' |
| 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 |
| (|) | * | + | , | - | . | / | 0 | 1 |
| 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; |
| 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 |
| < | = | > | ? | @ | A | B | C | D | E |
| 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 |
| F | G | H | I | J | K | L | M | N | O |
| 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 |
| P | Q | R | S | T | U | V | W | X | Y |
| 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 |
| Z | [| \ |] | ^ | _ | ` | a | b | c |
| 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 |
| d | e | f | g | h | i | j | k | l | m |
| 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 |
| n | o | p | q | r | s | t | u | v | w |
| 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 | 129 |
| x | y | z | { | | } | ~ | | Ç | ü |

Table 12. Graphics Printer Character Set 2 (continued)

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 |
| é | â | ä | à | â | ç | ê | ë | è | ï |
| 140 | 141 | 142 | 143 | 144 | 145 | 146 | 147 | 148 | 149 |
| î | ì | Ä | Â | É | æ | Æ | ô | ö | ò |
| 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 |
| û | ù | ÿ | ö | ü | ç | £ | ¥ | ₣ | ₹ |
| 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 |
| á | í | ó | ú | ñ | Ñ | ä | ü | ¿ | — |
| 170 | 171 | 172 | 173 | 174 | 175 | 176 | 177 | 178 | 179 |
| ½ | ¼ | ⅓ | ⅔ | ⅕ | ⅙ | ⅚ | ⅛ | ⅜ | ⅞ |
| 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 |
| ¡ | ¢ | £ | ¥ | ₣ | ₹ | ₱ | ₲ | ₳ | ₴ |
| 190 | 191 | 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 |
| ₵ | ₶ | ₷ | ₸ | ₹ | ₺ | ₻ | ₼ | ₽ | ₾ |
| 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 | 208 | 209 |
| ₿ | ₾ | ₿ | ₾ | ₿ | ₾ | ₿ | ₾ | ₿ | ₾ |
| 210 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 | 219 |
| ₿ | ₾ | ₿ | ₾ | ₿ | ₾ | ₿ | ₾ | ₿ | ₾ |
| 220 | 221 | 222 | 223 | 224 | 225 | 226 | 227 | 228 | 229 |
| ₿ | ₾ | ₿ | ₾ | ₿ | ₾ | ₿ | ₾ | ₿ | ₾ |
| 230 | 231 | 232 | 233 | 234 | 235 | 236 | 237 | 238 | 239 |
| μ | τ | ϑ | Θ | Ω | δ | ∞ | ∅ | ε | ∩ |
| 240 | 241 | 242 | 243 | 244 | 245 | 246 | 247 | 248 | 249 |
| ≡ | ± | ≥ | ≤ | ∫ | ∫ | ÷ | ≈ | ° | ■ |
| 250 | 251 | 252 | 253 | 254 | 255 | | | | |
| - | √ | ∩ | 2 | ■ | SP | | | | |

5 1/4-Inch Diskette Drive Adapter

The System Unit has space and power for one or two 5-1/4" Diskette Drives. The drives are soft sectored, single or double sided, with 40 tracks per side. They are Modified Frequency Modulation (MFM) coded in 512 byte sectors, giving a formatted capacity of 163,840 bytes per drive for single sided and 327,680 bytes per drive for double sided. They have a track to track access time of 8 ms and a motor start time of 500 ms.

The 5-1/4" Diskette Drive Adapter fits in one of the System Board's five System Expansion Slots. It attaches to the two drives via an internal daisy chained flat cable which connects to one end of the drive adapter. The adapter has a second connector on the other end which extends through the rear panel of the System Unit. This connector contains the signals for two additional external drives, thus the 5-1/4" Diskette Drive Adapter is capable of attaching four 5-1/4" drives, two internal, and two external.

The adapter is designed for double density MFM coded drives and uses write precompensation with an analog phase locked loop for clock and data recovery. The adapter is a general purpose device using the NEC μ PD765 compatible controller. Thus the drive parameters are programmable. In addition, the attachment supports the drive's write protect feature.

The adapter is buffered on the I/O bus and uses the System Board direct memory access (DMA) for record data transfers. An interrupt level is also used to indicate operation complete and status condition requiring processor attention.

In general, the 5-1/4" Diskette Drive Adapter presents a high-level command interface to software I/O drivers. A block diagram of the 5-1/4" Diskette Drive Adapter is on the following page.

5-1/4" Diskette Drive Adapter Block Diagram

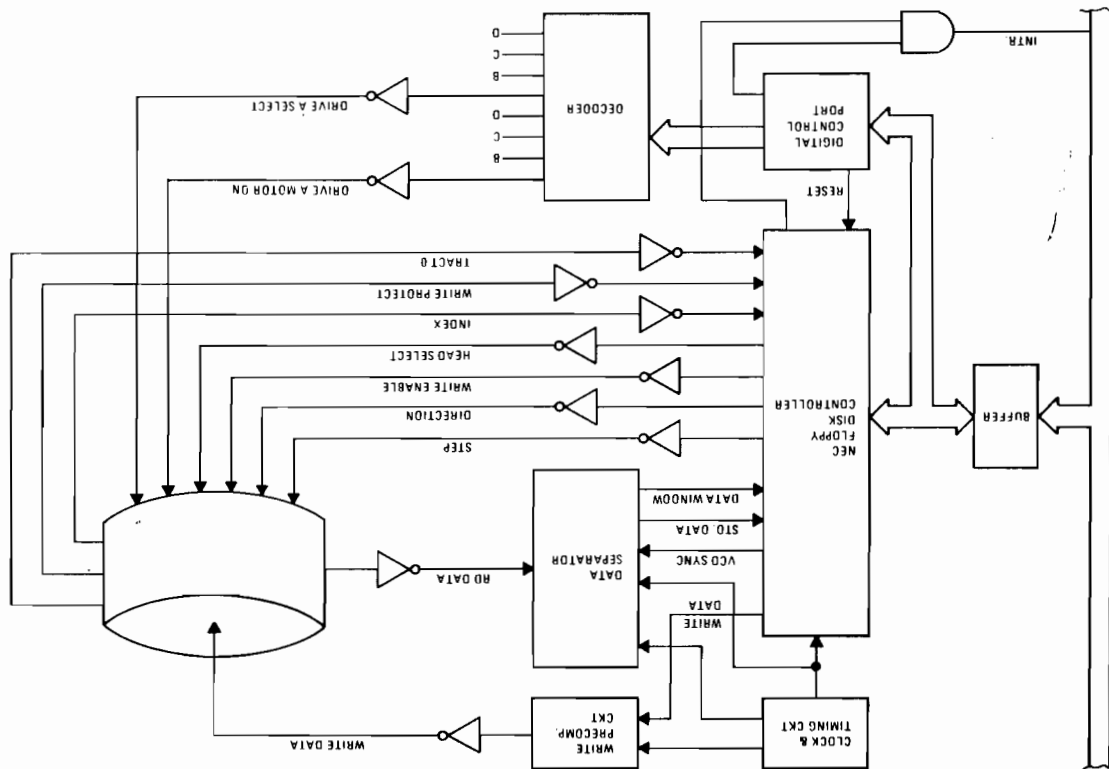


Figure 19. 5 1/4" DISKETTE DRIVE ADAPTER BLOCK DIAGRAM

Functional Description

From a programming point of view, this attachment consists of an 8-bit digital output register in parallel with a NEC μ PD765 or equivalent Floppy Disk Controller (FDC).

In the following description, drives numbers 0-3 are equivalent to drives A-D respectively.

Digital Output Register (DOR)

The Digital Output Register (DOR) is an output only register used to control drive motors, drive selection, and feature enable. All bits are cleared by the I/O interface reset line. The bits have the following functions:

Bits 0 and 1 These bits are decoded by the hardware to select one drive if its motor is on:

| Bit | 1 | 0 | Drive |
|-----|---|---|-------|
| 0 | 0 | 0 | A |
| 0 | 1 | 1 | B |
| 1 | 0 | 2 | C |
| 1 | 1 | 3 | D |

Bit 2 The FDC is held reset when this bit is clear. It must be set by the program to enable the FDC.

Bit 3 This bit allows the FDC interrupt and DMA requests to be gated onto the I/O interface. If this bit is cleared, the interrupt and DMA request I/O interface drivers are disabled.

Bits 4,5,6, and 7 These bits control respectively the motors of drives 0,1,2,A,B,C, and 3,D. If a bit is clear, the associated motor is off, and the drive cannot be selected.

Floppy Disk Controller (FDC)

The following is a brief summary of the registers and commands implemented by the FDC.

The FDC contains two registers which may be accessed by the main system processor; a Status Register and a Data Register. The 8-bit Main Status Register contains the status information of the FDC, and may be accessed at any time. The 8-bit Data Register (actually consisting of several registers in a stack with only one register presented to the data bus at a time) stores data, commands, parameters, and FDD status information. Data bytes are read out of, or written into, the Data Register in order to

program or obtain the results after a particular command. The Main Status Register may only be read and is used to facilitate the transfer of data between the processor and FDC.

The bits in the Main Status Register are defined as follows:

| Bit Number | Name | Symbol | Description |
|------------|--------------------|--------|---|
| DB0 FDD | FDD A Busy | DAB | FDD number is in the Seek mode. |
| DB1 | FDD B Busy | DBB | FDD number 1 is in the Seek mode. |
| DB2 | FDD C Busy | DCB | FDD number 2 is in the Seek mode. |
| DB3 | FDD D Busy | DDB | FDD number 3 is in the Seek mode. |
| DB4 | FDC Busy | CB | A read or write command is in process. |
| DB5 | Non-DMA Mode | NDM | The FDC is in the non-DMA mode. |
| DB6 | Data Input/ | DIO | Indicates direction of data transfer between FDC and Processor. If DIO = "1", then transfer is from FDC Data Register to the Processor. If DIO = "0", then transfer is from the Processor to FDC Data Register. |
| DB7 | Request for Master | RQM | Indicates Data Register is ready to send or receive data to or from the Processor. Both bits DIO and RQM should be used to perform the handshaking functions of "ready" and "direction" to the processor. |

The FDC is capable of performing 15 different commands. Each command is initiated by a multi-byte transfer from the processor, and the result after execution of the command may also be a multi-byte transfer back to the processor. Because of this multi-byte interchange of information between the FDC and the processor, it is convenient to consider each command as consisting of three phases:

Command Phase

The FDC receives all information required to perform a particular operation from the processor.

Execution Phase

The FDC performs the operation it was instructed to do.

Result Phase

After completion of the operation, status and other housekeeping information are made available to the processor.

Programming Considerations

Table 13. Symbol Description

The following tables define the symbols used in the command summary which follows.

| SYMBOL | NAME | DESCRIPTION |
|--------|---------------------|---|
| A0 | Address Line 0 | A0 controls selection of Main Status Register (A0 = 0) or Data Register (A0 = 1). |
| C | Cylinder Number | C stands for the current/selected Cylinder (track) number of the medium. |
| D | Data | D stands for the data pattern which is going to be written into a Sector. |
| D7-D0 | Data Bus | 8-bit Data Bus, where D7 stands for a most significant bit, and D0 stands for a least significant bit. |
| DTL | Data Length | When N is defined as 0, DTL stands for the data length which users are going to read out or write into the Sector. |
| EOT | End of Track | EOT stands for the final Sector number on a Cylinder. |
| GPL | Gap Length | GPL stands for the length of Gap 3 (spacing between Sectors excluding VCO Sync. Field). |
| H | Head Address | H stands for head number 0 or 1, as specified in ID field. |
| HD | Head | HD stands for a selected head number 0 or 1. (H = HD in all command words.) |
| HLT | Head Load Time | HLT stands for the head load time in the FDD (4 to 512 ms in 4 ms increments). |
| HUT | Head Unload Time | HUT stands for the head unload time after a read or write operation has occurred (0 to 480 ms in 32 ms increments.) |
| MF | FM or MFM Mode | If MF is low, FM mode is selected, and if it is high, MFM mode is selected only if MFM is implemented. |
| MT | Multi-Track | If MT is high, a multi-track operation is to be performed. (A cylinder under both HD0 and HD1 will be read or written.) |
| N | Number | N stands for the number of data bytes written in a Sector. |
| NCN | New Cylinder Number | NCN stands for a new Cylinder number, which is going to be reached as a result of the Seek operation. Desired position of Head. |

Table 13. Symbol Descriptions (continued)

| SYMBOL | NAME | DESCRIPTION |
|------------------------------|--|---|
| ND | Non-DMA Mode | ND stands for operation in the Non-DMA Mode. |
| PCN | Present Cylinder Number | PCN stands for Cylinder number at the completion of SENSE INTERRUPT STATUS Command, indicating the position of the Head at present time. |
| R | Record | R stands for the Sector number, which will be read or written. |
| R/W | Read/Write | R/W stands for either Read (R) or Write (W) signal. |
| SC | Sector | SC indicates the number of Sectors per Cylinder. |
| SK | Skip | SK stands for Skip Deleted Data Address Mark. |
| SRT | Step Rate Time | SRT stands for the Stepping Rate for the FDD. (2 to 32 ms in 2 ms increments.) |
| ST 0 ST 1 ST 2 ST 3 | Status 0 Status 1 Status 2 Status 3 | ST 0-3 stand for one of four registers which store the status information after a command has been executed. This information is available during the result phase after command execution. These registers should not be confused with the main status register (selected by A0 = 0). ST 0-3 may be read only after a command has been executed and contain information relevant to that particular command. |
| STP | Scan Test | During a Scan operation, if STP = 1, the data in contiguous sectors is compared byte by byte with data sent from the processor (or DMA), and if STP = 2, then alternate sectors are read and compared. |
| US0, US1 | Unit Select | US stands for a selected drive number encoded the same as bits 0 and 1 of the digital register (DOR) p 2-91 |

Command Summary

0 indicates 'logical 0' for that bit, 1 means 'logical 1', X means 'don't care'.

| PHASE | R/W | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | REMARKS |
|-----------|-----|----|----|----|----|----|----|-----|-----|---|
| Command | W | MT | MF | SK | 0 | 0 | 1 | 1 | 0 | Command Codes Sector ID information prior to Command execution |
| Execution | W | X | X | X | X | X | HD | US1 | US0 | |
| Result | W | | | | | | | | | Data-transfer between the FDD and main-system Status information after Command execution Sector ID information after Command execution |
| Command | W | MT | MF | SK | 0 | 1 | 1 | 0 | 0 | Command Codes Sector ID information prior to Command execution |
| Execution | W | X | X | X | X | X | HD | US1 | US0 | |
| Result | W | | | | | | | | | Data-transfer between the FDD and main-system Status information after command execution Sector ID information after command execution |
| Command | W | MT | MF | SK | 0 | 0 | 1 | 0 | 1 | Command Codes Sector ID information prior to command execution |
| Execution | W | X | X | X | X | X | HD | US1 | US0 | |
| Result | W | | | | | | | | | Data-transfer between the main-system and FDD Status information after command execution Sector ID information after command execution |

Command Summary (continued)

| PHASE | R/W | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | REMARKS |
|-----------|-----|----|----|----|----|----|----|-----|-----|---|
| Command | W | MT | MF | 0 | 0 | 1 | 0 | 0 | 1 | Command Codes Sector ID information prior to command execution |
| Execution | W | X | X | X | X | X | HD | US1 | US0 | |
| Result | W | | | | | | | | | Data-transfer between FDD and main-system Status ID information after common execution Sector ID information after command execution |
| Command | W | 0 | MF | SK | 0 | 0 | 0 | 1 | 0 | Command Codes Sector ID information prior to command execution |
| Execution | W | X | X | X | X | X | HD | US1 | US0 | |
| Result | W | | | | | | | | | Data-transfer between the FDD and main-system. FDC reads all of cylinders contents from index hole to EDT. Status information after command execution Sector ID information after command execution |
| Command | W | 0 | MF | SK | 0 | 0 | 0 | 1 | 0 | Command Codes Sector ID information prior to command execution |
| Execution | W | X | X | X | X | X | HD | US1 | US0 | |
| Result | W | | | | | | | | | Data-transfer between the FDD and main-system. FDC reads all of cylinders contents from index hole to EDT. Status information after command execution Sector ID information after command execution |
| Command | W | 0 | MF | 0 | 0 | 0 | 1 | 0 | 0 | Command Codes Sector ID information prior to command execution |
| Execution | W | X | X | X | X | X | HD | US1 | US0 | |
| Result | W | | | | | | | | | Data-transfer between the FDD and main-system. FDC reads all of cylinders contents from index hole to EDT. Status information after command execution Sector ID information after command execution |
| Command | W | 0 | MF | 0 | 0 | 0 | 1 | 0 | 0 | Command Codes Sector ID information prior to command execution |
| Execution | W | X | X | X | X | X | HD | US1 | US0 | |
| Result | W | | | | | | | | | Data-transfer between the FDD and main-system. FDC reads all of cylinders contents from index hole to EDT. Status information after command execution Sector ID information after command execution |

Command Summary (continued)

| PHASE | R/W | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | REMARKS |
|-----------|-----|----|----|----|------|----|----|-----|-----|--|
| Command | W | 0 | MF | 0 | 0 | 1 | 1 | 0 | 0 | Command Codes |
| Execution | W | X | X | X | X | X | HD | US1 | US0 | Bytes/Sector |
| Result | W | | | | N | | | | | Sector/Track |
| | W | | | | SC | | | | | Gap 3 |
| | W | | | | GPL | | | | | filler byte |
| | W | | | | D | | | | | FDC formats an entire cylinder |
| Result | R | | | | ST 0 | | | | | Status information after command execution |
| | R | | | | ST 1 | | | | | In this case, the ID information has no meaning |
| | R | | | | ST 2 | | | | | |
| | R | | | | C | | | | | |
| | R | | | | H | | | | | |
| | R | | | | H | | | | | |
| | R | | | | N | | | | | |
| Command | W | MT | MF | SK | 1 | 0 | 0 | 0 | 1 | Command Codes |
| Execution | W | X | X | X | X | X | HD | US1 | US0 | Sector ID information prior to command execution |
| Result | W | | | | C | | | | | Data compared between the FDD and main-system Status information after command execution |
| | W | | | | H | | | | | Sector ID information |
| | W | | | | R | | | | | |
| | W | | | | N | | | | | |
| | W | | | | EOT | | | | | |
| | W | | | | GPL | | | | | |
| | W | | | | STP | | | | | |
| Result | R | | | | ST 0 | | | | | Command Codes |
| | R | | | | ST 1 | | | | | Sector ID information prior to command execution |
| | R | | | | ST 2 | | | | | Data compared between the FDD and main-system Status information after command execution |
| | R | | | | C | | | | | Sector ID information |
| | R | | | | H | | | | | |
| | R | | | | H | | | | | |
| | R | | | | N | | | | | |
| Command | W | MT | MF | SK | 1 | 1 | 0 | 0 | 1 | Command Codes |
| Execution | W | X | X | X | X | X | HD | US1 | US0 | Sector ID information prior to command execution |
| Result | W | | | | C | | | | | Data compared between the FDD and main-system Status information after command execution |
| | W | | | | H | | | | | Sector ID information |
| | W | | | | R | | | | | |
| | W | | | | N | | | | | |
| | W | | | | EOT | | | | | |
| | W | | | | GPL | | | | | |
| | W | | | | STP | | | | | |
| Result | R | | | | ST 0 | | | | | Command Codes |
| | R | | | | ST 1 | | | | | Sector ID information prior to command execution |
| | R | | | | ST 2 | | | | | Data compared between the FDD and main-system Status information after command execution |
| | R | | | | C | | | | | Sector ID information |
| | R | | | | H | | | | | |
| | R | | | | H | | | | | |
| | R | | | | N | | | | | |

Command Summary (continued)

| PHASE | R/W | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | REMARKS |
|-----------------|-----|----|----|----|------------------------|----|----|-----|-----|--|
| Command | W | MT | MF | SK | 1 | 1 | 1 | 0 | 1 | Command Codes |
| Execution | W | X | X | X | X | X | HD | US1 | US0 | Sector ID information prior to command execution |
| Result | W | | | | C | | | | | Data compared between the FDD and main-system Status information after command execution |
| | W | | | | H | | | | | Sector ID information |
| | W | | | | R | | | | | |
| | W | | | | N | | | | | |
| | W | | | | EOT | | | | | |
| | W | | | | GPL | | | | | |
| | W | | | | STP | | | | | |
| Result | R | | | | ST 0 | | | | | Command Codes |
| | R | | | | ST 1 | | | | | Sector ID information prior to command execution |
| | R | | | | ST 2 | | | | | Data compared between the FDD and main-system Status information after command execution |
| | R | | | | C | | | | | Sector ID information |
| | R | | | | H | | | | | |
| | R | | | | H | | | | | |
| | R | | | | N | | | | | |
| Command | W | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | Command Codes |
| Execution | W | X | X | X | X | X | 0 | 1 | 1 | Head retracted to track 0 |
| No Result Phase | W | | | | | | | | | |
| Command | W | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | Command Codes |
| Result | R | | | | ST 0 | | | | | Status information at the end of seek operation about the FDC |
| | R | | | | PCN | | | | | |
| Command | W | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | Command Codes |
| No Result Phase | W | | | | —SRT— | | | | | |
| | W | | | | —HLT— | | | | | |
| | | | | | | | | | | ND |
| Command | W | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | Command Codes |
| Result | R | | | | SENSE INTERRUPT STATUS | | | | | |
| | R | | | | ST 0 | | | | | |
| | R | | | | PCN | | | | | |
| Command | W | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | Command Codes |
| No Result Phase | W | | | | —SRT— | | | | | |
| | W | | | | —HLT— | | | | | |
| | | | | | | | | | | ND |
| Command | W | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | Command Codes |
| Result | R | | | | SENSE DRIVE STATUS | | | | | |
| | R | | | | ST 3 | | | | | |
| Command | W | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | Command Codes |
| Execution | W | X | X | X | X | X | HD | US1 | US0 | Status information about FDD |
| No Result Phase | W | | | | | | | | | |
| Command | W | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | Command Codes |
| Result | R | | | | SEEK | | | | | Head is positioned over proper cylinder on diskette |
| | R | | | | NCN | | | | | |
| Command | W | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | Command Codes |
| Execution | W | X | X | X | X | X | HD | US1 | US0 | Status information about FDD |
| No Result Phase | W | | | | | | | | | |
| Command | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Command Codes |
| Result | R | | | | INVALID | | | | | Invalid command codes (NoOp - FDC goes into standby state) ST 0 = 80 |
| | R | | | | Invalid Codes | | | | | |
| | R | | | | ST 0 | | | | | |

Command Status Registers

Table 14. Status Register 0

| NO. | BIT | | DESCRIPTION |
|----------|--------------------------------|--------------|--|
| | NAME | SYMBOL | |
| D7 | Interrupt Code | IC | D7 = 0 and D6 = 0 Normal termination of command, (NT), Command was completed and properly executed. D7 = 0 and D6 = 1 Abnormal termination of command, (AT). Execution of command was started, but was not successfully completed. D7 = 1 and D6 = 0 Invalid command issue (IC). Command which was issued was never started. D7 = 1 and D6 = 1 Abnormal termination because during command execution the ready signal from FDD changed state. |
| D5 | Seek End | SE | When the FDC completes the Seek command, this flag is set to 1 (high). |
| D4 | Equipment Check | EC | If a fault signal is received from the FDD, or if the track 0 signal fails to occur after 77 step pulses (recalibrate command) then this flag is set. |
| D3 | Not Ready | NR | When the FDD is in the not-ready state and a read or write command is issued, this flag is set. If a read or write command is issued to side 1 of a single sided drive, then this flag is set. |
| D2 | Head Address | HD | This flag is used to indicate the state of the head at interrupt. |
| D1 D0 | Unit Select 1 Unit Select 0 | US 1 US 0 | These flags are used to indicate a Drive unit Number at interrupt. |

Table 15. Status Register 1

| NO. | BIT | | DESCRIPTION |
|-----|----------------------|--------|---|
| | NAME | SYMBOL | |
| D7 | End of Cylinder | EN | When the FDC tries to access a sector beyond the final sector of a cylinder, this flag is set. |
| D6 | — | — | Not used. This bit is always 0 (low). |
| D5 | Data Error | DE | When the FDC detects a CRC error in either the ID field or the data field, this flag is set. |
| D4 | Over Run | OR | If the FDC is not serviced by the main-systems during data transfers within a certain time interval, this flag is set. |
| D3 | — | — | Not used. This bit is always 0 (low). |
| D2 | No Data | ND | During Execution of a Read Data, Write Deleted Data, or Scan command, if the FDC cannot find the sector specified in the ID register, this flag is set. During execution of the Read ID command, if the FDC cannot read the ID field without an error, then this flag is set. During the execution of the Read-a-Cylinder command, if the starting sector cannot be found, then this flag is set. |
| D1 | Not Writable | NW | During Execution of a Write Data, Write Deleted Data, or Format a Cylinder command, if the FDC detects a write protect signal from the FDD, then this flag is set. |
| D0 | Missing Address Mark | MA | If the FDC cannot detect the ID Address Mark, this flag is set. Also at the same time, the MD (Missing Address Mark in Data Field) of Status Register 2 is set. |

Table 16. Status Register 2

| NO. | BIT | | DESCRIPTION |
|-----|------------------------------------|--------|---|
| | NAME | SYMBOL | |
| D7 | — | — | Not Used. This bit is always 0 (low). |
| D6 | Control Mark | CM | During execution of the Read Data or Scan command, if the FDC encounters a sector which contains a Deleted Data Address Mark, this flag is set. |
| D5 | Data Error in Data Field | DD | If the FDC detects a CRC error in the data then this flag is set. |
| D4 | Wrong Cylinder | WC | This bit is related with the ND bit, and when the contents of C on the medium are different from that stored in the ID Register, this flag is set. |
| D3 | Scan Equal Hit | SH | During execution of the Scan command, if the condition of "equal" is satisfied, this flag is set. |
| D2 | Scan Not Satisfied | SN | During execution of the Scan command, if the FDC cannot find a sector on the cylinder which meets the condition, then this flag is set. |
| D1 | Bad Cylinder | BC | This bit is related with the ND bit, and when the contents of C on the medium are different from that stored in the ID Register, and the content of C is FF, then this flag is set. |
| D0 | Missing Address Mark in Data Field | MD | When data is read from the medium, if the FDC cannot find a Data Address Mark or Deleted Data Address Mark, then this flag is set. |

Table 17. Status Register 3

| NO. | BIT | | DESCRIPTION |
|-----|-----------------|--------|---|
| | NAME | SYMBOL | |
| D7 | Fault | FT | This bit is used to indicate the status of the Fault signal from the FDD. |
| D6 | Write Protected | WP | This bit is used to indicate the status of the Write Protected signal from the FDD. |
| D5 | Ready | RY | This bit is used to indicate the status of the Ready signal from the FDD. |
| D4 | Track 0 | T0 | This bit is used to indicate the status of the Track 0 signal from the FDD. |
| D3 | Two Side | TS | This bit is used to indicate the status of the Two Side signal from the FDD. |
| D2 | Head Address | HD | This bit is used to indicate the status of Side Select signal to the FDD. |
| D1 | Unit Select 1 | US 1 | This bit is used to indicate the status of the Unit Select 1 signal to the FDD. |
| D0 | Unit Select 0 | US 0 | This bit is used to indicate the status of the Unit Select 0 signal to the FDD. |

Programming Summary

DPC Registers (Ports)

FDC Data Reg

I/O Address 3F5

FDC Main Status Reg

I/O Address 3F4

Digital Output Reg

I/O Address 3F2

Bit 0 Drive 00: DR #A 10: DR #C

1 Select 01: DR #B 11: DR #D

2 Not FDC Reset

3 Enable INT & DMA Requests

4 Drive A Motor Enable

5 Drive B Motor Enable

6 Drive C Motor Enable

7 Drive D Motor Enable

All bits cleared with channel reset.

Interrupt 6**DMA 2****100 Disk Format**

1 Head, 45 cylinders, 8 sectors/TRK, 512 bytes/sector, MFH.

FDC Constants

N: H'02', SC: 08, HUT: F, SRT: C, GPL FORMAT: H'05',
GPL RD/WR: 2A, HLT: 01, (8ms track-track)

Drive Constants

HD Load 35 ms
HD Settle 25 ms
Motor Start 500 ms

Comments

1. Head loads with drive select, wait HD Load time before RD/WR.
2. Following access, wait HD Settle time before RD/WR.
3. Drive motors should be off when not in use. Only A or B and C or D may run simultaneously. Wait Motor Start time before RD/WR.
4. Motor must be on for drive to be selected.
5. Data Errors can occur while using a Home Television as the system display. Locating the TV too close to the diskette area can cause this to occur. To correct the problem, move the TV away from, or to the opposite side of the System Unit.

System I/O Channel Interface

All signals are TTL compatible:

MPUL 5.5 Vdc
LPUL 2.7 Vdc
MPDL 0.5 Vdc
LPDL -0.5 Vdc

The following lines are used by this adapter.

+D0-7 (Bidirectional, Load: 1 74LS; Driver: 74LS 3-state)

These eight lines form a bus by which all commands, status, and data are transferred. Bit 0 is the low-order bit.

+A0-9

(Adapter Input, Load: 1 74LS)

These ten lines form an address bus by which a register is selected to receive or supply the byte transferred via lines D0-7. Bit 0 is the low-order bit.

+AEN

(Adapter Input, Load: 1 74LS)

The content of lines A0-9 is ignored if this line is active.

-IOW

(Adapter Input, Load: 1 74LS)

The content of lines D0-7 is stored in the register addressed by lines A0-9 or DACK2 at the trailing edge of this signal.

-IOR

(Adapter Input, Load: 1 74LS)

The content of the register addressed by lines A0-9 or DACK2 is gated onto lines D0-7 when this line is active.

-DACK2

(Adapter Input, Load: 2 74LS)

This line active negates output DRQ2, selects the FDC data register as the source/destination of bus D0-7, and indirectly gates T/C to IRQ6.

+T/C

(Adapter Input, Load: 4 74LS)

This line and DACK2 active indicates that the byte of data for which the DMA count was initialized is now being transferred.

+RESET

(Adapter Input, Load: 1 74LS)

An up level aborts any operation in process and clears the Digital Output Register (DOR).

+DRQ2

(Adapter Output, Driver: 74LS 3-state)

This line is made active when the attachment is ready to transfer a byte of data to or from main storage. The line is made inactive by DACK2 becoming active or an I/O read of the FDC data register.

+IRQ6

(Adapter Output, Driver: 74LS 3-state)

This line is made active when the FDC has completed an operation. It results in an interrupt to a routine which should examine the FDC result bytes to reset the line and determine the ending condition.

Drive A & B Interface

All signals are TTL compatible:

MPUL 5.5 Vdc
LPUL 2.4 Vdc
MPDL 0.4 Vdc
LPDL -0.5 Vdc

All adapter outputs are driven by open-collector gates. The drive(s) must provide termination networks to Vcc (except Motor Enable 1 which has a 2K ohm resistor to Vcc).

Each adapter input is terminated with a 150 ohm resistor to Vcc.

Adapter Outputs

-Drive Select A&B (Driver: 7438)

These two lines are used by drives A&B to degate all drivers to the adapter and receivers from the attachment (except Motor Enable) when the line associated with a drive is inactive.

-Motor Enable A&B (Driver: 7438)

The drive associated with each of these lines must control its spindle motor such that it starts when the line becomes active and stops when the line becomes inactive.

-Step (Driver: 7438)

The selected drive moves the read/write head one cylinder in or out per the direction line for each pulse present on this line.

-Direction (Driver: 7438)

For each recognized pulse of the step line the read/write head moves one cylinder toward the spindle if this line is active, and away from the spindle if inactive.

-Head Select (Driver: 7438)

Head 1 (upper head) will be selected when this line is active (low).

-Write Data (Driver: 7438)

For each inactive to active transition of this line while Write Enable is active, the selected drive causes a flux change to be stored on the disk.

Adapter Inputs

-Write Enable

(Driver: 7438)

The drive disables write current in the head unless this line is active.

-Index

The selected drive supplies one pulse per disk revolution on this line.

-Write Protect

The selected drive makes this line active if a write protected diskette is mounted in the drive.

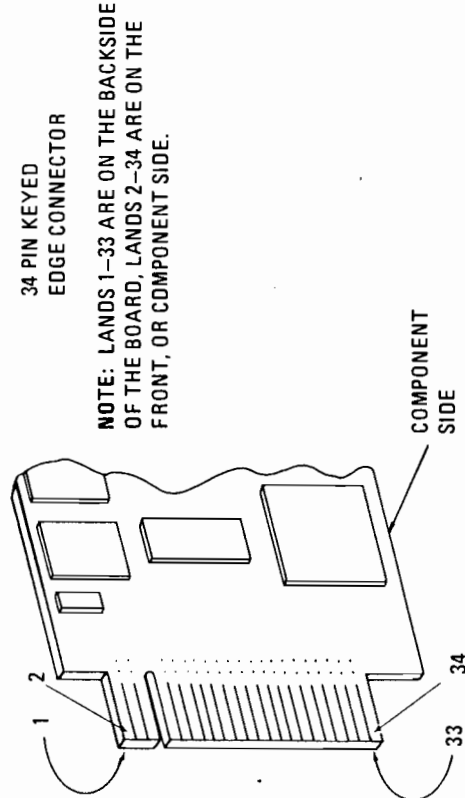
-Track 0

The selected drive makes this line active if the read/write head is over track 0.

-Read Data

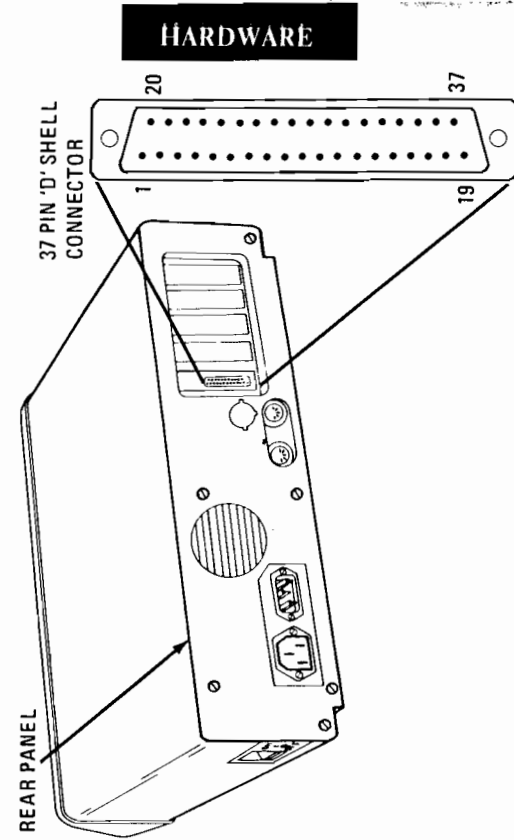
The selected drive supplies a pulse on this line for each flux change encountered on the disk.

5-1/4" Diskette Drive Adapter Internal Interface Specifications



| AT STANDARD TTL LEVELS | | Land No. |
|-------------------------------|---------------------------|----------|
| IBM 5 1/4" Diskette Drives | Ground-Odd Numbers | 1-33 |
| | Unused | 2,4,6 |
| | Index | 8 |
| | Motor Enable A | 10 |
| | Drive Select B | 12 |
| | Drive Select A | 14 |
| | Motor Enable B | 16 |
| | Direction (Stepper Motor) | 18 |
| | Step Pulse | 20 |
| | Write Data | 22 |
| | Write Enable | 24 |
| | Track 0 | 26 |
| | Write Protect | 28 |
| | Read Data | 30 |
| | Select Head 1 | 32 |
| | Unused | 34 |
| 5 1/4" Diskette Drive Adapter | | |

5-1/4" Diskette Drive Adapter External Interface Specifications



HARDWARE

| AT STANDARD TTL LEVELS | | Pin no. |
|-------------------------------|---------------------------|---------|
| External Drives | Unused | 1-5 |
| | Index | 6 |
| | Motor Enable C | 7 |
| | Drive Select D | 8 |
| | Drive Select C | 9 |
| | Motor Enable D | 10 |
| | Direction (Stepper Motor) | 11 |
| | Step Pulse | 12 |
| | Select Head 1 | 13 |
| | Write Enable | 14 |
| | Track 0 | 15 |
| | Write Protect | 16 |
| | Read Data | 17 |
| | Write Data | 18 |
| | Ground | 20-37 |
| 5 1/4" Diskette Drive Adapter | | |

5-1/4" Diskette Drive

The IBM 5-1/4" Diskette Drive is a single or double sided, double density, 40 track per side unit. The Diskette Drive has a formatted capacity of 163,840 bytes for single sided and 327,680 bytes for double sided, and is capable of reading and recording digital data using Modified Frequency Modulation (MFM) methods. User access for diskette loading is provided by way of a slot located at the front of the unit.

The Diskette Drive is fully self-contained and requires no operator intervention during normal operation. The Drive consists of a spindle drive system, a head positioning system, and read/write/erase system.

When the front latch is opened, access is provided for the insertion of a diskette. The diskette is positioned in place by plastic guides, and the front latch. In/out location is ensured when the diskette is inserted until a back stop is encountered.

Closing the front latch activates the cone/clamp system resulting in centering of the diskette and clamping of the diskette to the drive hub. The drive hub is driven at a constant speed of 300 rpm by a servo controlled DC motor. In operation, the magnetic head is loaded into contact with the recording medium whenever the front latch is closed.

The magnetic head is positioned over the desired track by means of a 4-phase stepper motor/band assembly and its associated electronics. This positioner employs a one-step rotation to cause a 1-track linear movement. When a write-protected diskette is inserted into the Drive, the write-protect sensor disables the write electronics of the Drive and an appropriate signal is applied to the interface.

When performing a write operation, a 0.33 mm (0.013-in.) data track is recorded. This track is then tunnel erased to 0.30 mm (0.012-in.).

Data recovery electronics include a low-level read amplifier, differentiator, zero-crossing detector, and digitizing circuits. All data decoding is provided by the adapter card.

The Drive is also supplied with the following sensor systems:

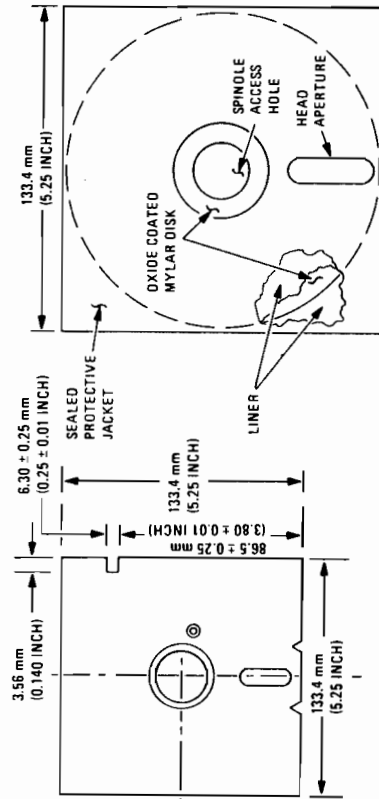
- (1) A track 00 switch which senses when the Head/Carriage assembly is positioned at Track 00.
- (2) The index sensor, which consists of a LED light source and phototransistor, is positioned so that when an index hole is detected, a digital signal is generated.

- (3) The write-protect sensor disables the Diskette Drive electronics whenever a write-protect tab is applied to the diskette.

For interface information, refer to the Diskette Drive Adapter section.

Diskettes

The IBM 5-1/4" Diskette Drive uses a standard 133.4 mm (5.25 in.) diskette. For programming considerations, single sided, double density, soft sectored diskettes are used for single sided drives and double sided, double density, soft sectored diskettes are used for double sided drives. The figure below is a simplified drawing of the diskette used with the Diskette Drive. This recording medium is a flexible magnetic disk enclosed in a protective jacket. The protected disk, free to rotate within the jacket, is continuously cleaned by the soft fabric lining of the jacket during normal operation. Read/Write erase head access is made through an opening in the jacket. Openings for the drive hub and diskette index hole are also provided.



RECORDING MEDIUM

Table 18. Mechanical and Electrical Specifications

| | |
|---|--|
| Media | Industry-compatible 5¼-inch diskette |
| Tracks per inch | 48 |
| Number of Tracks | (40) |
| Dimensions | |
| Height | 85.85 mm (3.38 inches) |
| Width | 149.10 mm (5.87 inches) |
| Depth | 203.2 mm (8.0 inches) |
| Weight | 2.04 Kg (4.5 lbs.) |
| Temperature (Exclusive of Media) | |
| Operating | 10°C to 44°C (50°F to 112°F) |
| Non-operating | -40°C to 60°C (-40°F to 140°F) |
| Relative Humidity (Exclusive of Media) | |
| Operating | 20% to 80% (Non-condensing) |
| Non-operating | 5% to 95% (Non-condensing) |
| Seek Time | 8 msec track to track |
| Head Setting Time | 25 msec (last track addressed) |
| Error Rate | 1 per 10 ⁹ (recoverable) 1 per 10 ¹² (non-recoverable) 1 per 10 ⁶ (seeks) |
| Head Life | 20,000 hours (normal use) |
| Media Life | 3.0 x 10 ⁶ passes per track |
| Disk Speed | 300 rpm ± 1.5% (long term) |
| Instantaneous Speed Variation | ± 3.0% |
| Start/Stop Time | 500 msec (maximum) |
| Transfer Rate | 250K bits/sec |
| Recording Mode | MFM |
| Power | +12 dc ± 0.6v 900 ma AVE. +5v dc ± 0.25 v, 600 ma AVE. |

Memory Expansion Options

Three Memory Expansion Options are offered for the IBM Personal Computer. They are the 32KB, the 64KB, and the 64/256KB Memory Expansion Options. The base 64/256KB Option comes standard with 64KB of memory. One, two, or three 64KB Memory Module Kits may be added, providing the 64/256KB Option with 128KB, 192KB, or 256KB of memory respectively. The Memory Expansion Options plug into any of the five System Expansion Slots on the System Board and are used to extend system memory beyond 64KB. A maximum of 64KB of memory may be installed on the System Board as modules without using any System Expansion Slots or Expansion Options.

An expansion option must be configured to reside at a sequential 32K or 64K memory address boundary within the system address space. This is done by setting DIP switches on the option.

The expansion options are designed with 250 ns access dynamic memory chips. The 32KB and the 64KB options both use 16K x 1 memory chips while the 64/256KB option uses 64K x 1 memory chips. On the 32KB and the 64/256KB cards, 16 pin industry standard parts are used. On the 64KB card, stacked modules are used resulting in a 32K x 1, 18 pin module. This allows the 32KB and 64KB cards to have approximately the same physical dimensions.

All expansion options are parity checked and if a parity error is detected, a latch is set and an I/O channel check line is activated, indicating an error to the processor.

In addition to the memory modules, the expansion options contain the following circuits: bus buffering, dynamic memory timing generation, address multiplexing, and card select decode logic.

Dynamic memory refresh timing and address generation are functions which are not performed on the expansion options but are done once on the System Board and made available in the I/O channel for all devices.

To allow the System to address 32KB, 64KB, 64/256KB Memory Expansion Options, refer to the system configuration switch settings on page 2-28.

Operating Characteristics

The System Board operates at a frequency of 4.77 Mhz, which results in a clock frequency of 210 ns.

Normally, five clock cycles are required for a bus cycle so that a 1.05 μ s memory cycle time is achieved. Memory write and memory read cycles both take five clock cycles, or 1.05 μ s.

General specifications for memory used on all cards are:

Access - 250 ns
Cycle - 410 ns

Memory Module Description

Both the 32KB and 64KB options contain 18 dynamic memory modules. The 32KB Memory Expansion Option utilizes 16K x 1 bit modules and the 64KB Memory Expansion Option utilizes 32K x 1 bit modules.

The 64/256KB card has four banks of 9 pluggable connectors. Each bank will accept a 64KB Memory Module Kit, consisting of 9 (64K x 1) modules. The kits must be installed sequentially into Banks 1, 2, and 3. The base 64/256KB card comes standard with the first bank of modules installed into Bank 0, providing 64KB of memory. One, two, or three 64KB Kits may be added, upgrading the card to 128KB, 192KB, or 256KB of memory.

The 16K x 1 and the 32K x 1 modules require three voltage levels: +5 Vdc, -5 Vdc, and +12 Vdc. The 64K x 1 modules only require one voltage level of +5 Vdc. All three memory modules require 128 refresh cycles every 2 msec. Absolute maximum access times are:

From RAS: 250 ns
From CAS: 160 to 165 ns

Table 19. Memory Module Pin Configuration

| PIN NO. | 16K X 1 BIT MODULE (Used on 32KB Card) | 32K X 1 BIT MODULE (Used on 64KB Card) | 64K X 1 BIT MODULE (Used on 64/256KB Card) |
|---------|---|---|---|
| 1 | -5V | -5V | N/C |
| 2 | Data In** | Data In** | Data In*** |
| 3 | -Write | -Write | -Write |
| 4 | -RAS | -RAS 0 | -RAS |
| 5 | A0 | -RAS 1 | A0 |
| 6 | A2 | A0 | A2 |
| 7 | A1 | A2 | A1 |
| 8 | +12V | A1 | +5VDC |
| 9 | +5V | +12V | A7 |
| 10 | A5 | +5V | A5 |
| 11 | A4 | A5 | A4 |
| 12 | A3 | A4 | A3 |
| 13 | A6 | A3 | A6 |
| 14 | Data Out** | A6 | Data Out*** |
| 15 | -CAS | Data Out** | -CAS |
| 16 | GND | -CAS 1 | GND |
| 17 | -* | -CAS 0 | . |
| 18 | -* | GND | . |

* 16K X 1 and 64K X 1 bit module has only 16 pins.

** Data In and Data Out are tied together (three state bus).

*** Data In and Data Out are tied together on Data Bits 0-7 (three state bus).

Switch-Configurable Start Address

Each card has a small DIP Module which contains eight switches. The switches are used to set the card start address as follows:

Table 20. DIP Module Start Address

| NO. | 32KB AND 64KB OPTIONS | 64/256KB OPTION |
|-----|-----------------------------|-----------------------|
| 1 | ON: A19=0; OFF: A19=1 | ON: A19=0; OFF: A19=1 |
| 2 | ON: A18=0; OFF: A18=1 | ON: A18=0; OFF: A18=1 |
| 3 | ON: A17=0; OFF: A17=1 | ON: A17=0; OFF: A17=1 |
| 4 | ON: A16=0; OFF: A16=1 | ON: A16=0; OFF: A16=1 |
| 5 | ON: A15=0; OFF: A15=1* | ON: Select 64KB |
| 6 | Not Used | ON: Select 128KB |
| 7 | Not Used | ON: Select 192KB |
| 8 | Used Only In 64KB RAM Card* | ON: Select 256KB |

* Switch No. 8 may be set on the 64KB Memory Expansion. Option to use only half the memory on the card (i.e., 32KB). If Switch No. 8 is ON, all 64KB is accessible. If Switch No. 8 is OFF, address bit A15 (as set by Switch No. 5) is used to determine which 32KB are accessible and the 64KB option behaves exactly like a 32KB option.

Game Control Adapter

HARDWARE

The Game Control Adapter allows the system to attach paddles and joysticks. Up to four paddles or two joysticks may be attached. In addition, four inputs for switches are provided. Paddle and joystick positions are determined by changing resistive values sent to the adapter. The adapter plus system software converts the present resistive value to a relative paddle or joystick position. On receipt of an output signal, four timing circuits are started. By determining the time required for the circuit to time out (a function of the resistance), the paddle position can be determined. This card could be used as a general purpose I/O card with four analog (resistive) inputs plus four digital input points. This card fits into any of the five System Board I/O slots. The game control interface cable attaches to the rear of the card which protrudes through the rear panel of the System Unit.

Game Control Adapter Block Diagram

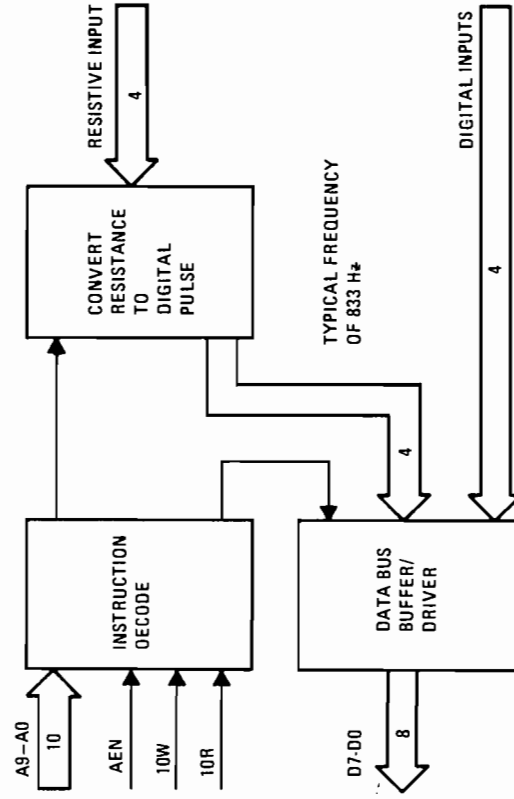


Figure 20. GAME CONTROL ADAPTER BLOCK DIAGRAM

Functional Description

Address Decode

The select on the Game Control Adapter is generated by two 74LS138's as an address decoder. AEN must be inactive while the address is 201 in order to generate the select. The select allows a write to fire the one-shots or a read to give the values of the trigger buttons and one-shot outputs.

Data Bus Buffer/Driver

The data bus is buffered by a 74LS244 buffer/driver. For an IN from address X'201', the Game Control Adapter will drive the data bus; at all other times the buffer is left in the high impedance state.

Trigger Buttons

The trigger button inputs are read via an IN from address X'201'. A trigger button is on each joystick/paddle. These values are seen on data bits 7 through 4 (see Software Interface sub-section). These buttons default to an open state and are read as "1". When a button is depressed, it is read as "0". Software should be aware that these buttons are NOT debounced in hardware.

Joystick Positions

The joystick position is indicated by a potentiometer for each coordinate. Each potentiometer has a range from 0 to 100K ohms that varies the time constant for each of the four one-shots. As this time constant is set at different values, the output of the one-shot will be of varying durations.

All four one-shots are fired at once by an OUT to address X'201'. All four one-shot outputs will go true after the fire pulse and will remain high for varying times depending on where each potentiometer is set.

These four one-shot outputs are read via an IN from address X'201' and are seen on data bits 3 through 0.

I/O Channel Description

| | |
|------------------|---|
| A9-A0: | Address lines 9 through 0 are used to address the Game Control Adapter. |
| D7-D0: | Data lines 7 through 0 are the data bus. |
| IOR, IOW: | I/O Read and I/O Write are used when reading from or writing to an adapter (IN, OUT). |
| AEN: | When active, the adapter must be inactive and the data bus driver inactive. |
| +5V: | Power for the Game Control Adapter. |
| GND: | Common ground. |
| A19-A10: | Unused |
| MEMR, MEMW: | Unused |
| DACK0-DACK3: | Unused |
| IRQ7-IRQ2: | Unused |
| DRQ3-DRQ1: | Unused |
| ALE, T/C: | Unused |
| CLK, OSC: | Unused |
| I/O CHCK: | Unused |
| I/O CH RDY: | Unused |
| HRQ I/O CH: | Unused |
| RESET DRV: | Unused |
| -5V, +12V, -12V: | Unused |

Interface Description

The Game Control Adapter has 8 input lines, 4 of which are digital inputs and 4 of which are resistive inputs. The inputs are read with one IN from address X'201'.

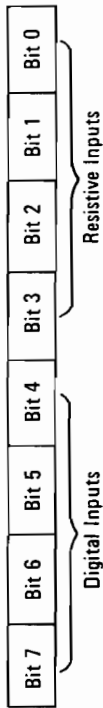
The 4 digital inputs each have a 1K ohm pullup resistor to +5V. With no drive on these inputs, a '1' is read. For a '0' reading, the inputs must be pulled to ground.

The 4 resistive inputs, measured to +5V, will be converted to a digital pulse with a duration proportional to the resistive load, according to the following equation:

$$\text{Time} = 24.2 \mu\text{sec} + 0.011 (r) \mu\text{sec}$$

The user must first begin the conversion by an OUT to address x'201'. An IN from address x'201' will show the digital pulse go high and remain high for the duration according to the resistance value. All four bits (Bit 3-Bit 0) function in the same manner, their digital pulse will all go high simultaneously and will reset independently according to the input resistance value.

Input from address x'201'



The typical input to the Game Control Adapter is a set of joysticks or game paddles.

The joysticks will typically have a set of two joysticks (A&B). These will have one or two buttons each with two variable resistances each, with a range from 0 to 100K ohms. One variable resistance will indicate the X coordinate and the other variable resistance will indicate the Y coordinate. This should be attached to give the following input data:

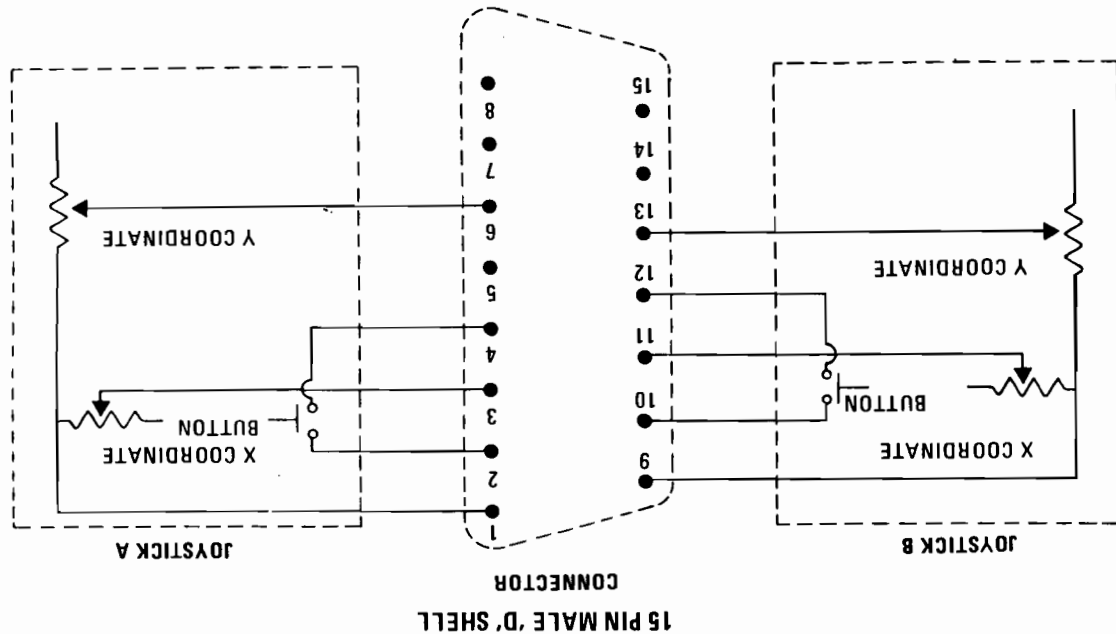
| | | | | | | | |
|-------------|-------------|-------------|-------------|------------|------------|------------|------------|
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| B-#2 Button | B-#1 Button | A-#2 Button | A-#1 Button | B-Y Coord. | B-X Coord. | A-Y Coord. | A-X Coord. |

The game paddles will have a set of two (A&B) or four (A,B,C, & D) paddles. These will have one button each and one variable resistance each, with a range from 0 to 100K ohms. This should be attached to give the following input data:

| | | | | | | | |
|----------|----------|----------|----------|----------|----------|----------|----------|
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| D Button | C Button | B Button | A Button | D Coord. | C Coord. | B Coord. | A Coord. |

A schematic diagram for attaching a set of game controllers is on page 2-135.

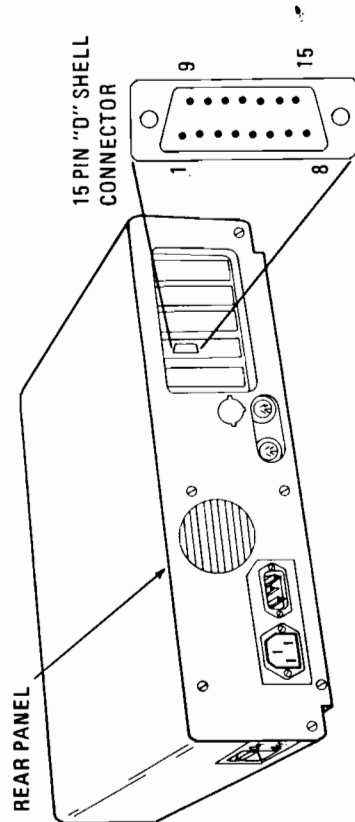
Joystick Schematic



NOTE: POTENTIOMETER FOR X & Y COORDINATES HAS A RANGE OF 0 TO 100KΩ. BUTTON IS NORMALLY OPEN, CLOSED WHEN DEPRESSED.

Figure 21. JOYSTICK SCHEMATIC

Game Controller Adapter (Analog Input) Connector Specifications



AT STANDARD TTL LEVELS:

| Voltage | AMP Pin No. |
|------------|-------------|
| + 5 Volts | 1 |
| Button 4 | 2 |
| Position 0 | 3 |
| Ground | 4 |
| Ground | 5 |
| Position 1 | 6 |
| Button 5 | 7 |
| + 5 Volts | 8 |
| + 5 Volts | 9 |
| Button 6 | 10 |
| Position 2 | 11 |
| Ground | 12 |
| Position 3 | 13 |
| Button 7 | 14 |
| + 5 Volts | 15 |

Game Control
Adapter

External
Devices

Asynchronous Communications Adapter

The Asynchronous Communications Adapter is a 4"H x 5"W card that plugs into a System Expansion Slot. All system control signals and voltage requirements are provided through a 2 x 31 position card edge tab. A jumper module is provided to select either RS-232-C or current loop operation.

The adapter is fully programmable and supports asynchronous communications only. It will add and remove start bits, stop bits, and parity bits. A programmable baud rate generator allows operation from 50 baud to 9600 baud. Five, six, seven or eight bit characters with 1, 1-1/2, or 2 stop bits are supported. A fully prioritized interrupt system controls transmit, receive, error, line status and data set interrupts. Diagnostic capabilities provide loopback functions of transmit/receive and input/output signals.

Figure (22) is a block diagram of the Asynchronous Communications Adapter.

The heart of the adapter is a INS8250 LSI chip or functional equivalent. The following is a summary of the 8250's key features:

- Adds or Deletes Standard Asynchronous Communication Bits (Start, Stop, and Parity) to or from Serial Data Stream.
- Full Double Buffering Eliminates Need for Precise Synchronization.
- Independently Controlled Transmit, Receive, Line Status, and Data Set Interrupts.
- Programmable Baud Rate Generator Allows Division of Any Input Clock by 1 to ($2^{16}-1$) and Generates the Internal 16x Clock.
- Independent Receiver Clock Input.
- MODEM Control Functions – Clear to Send (CTS), Request to Send (RTS), Data Set Ready (DSR), Data Terminal Ready (DTR), Ring Indicator (RI), and Carrier Detect (CD).
- Fully Programmable Serial-Interface Characteristics
 - 5-, 6-, 7-, or 8-Bit Characters
 - Even, Odd, or No-Parity Bit Generation and Detection
 - 1-, 1-1/2, or 2-Stop Bit Generation
 - Baud Rate Generation (DC to 9600 Baud)

- False Start Bit Detection.
- Complete Status Reporting Capabilities.
- Line Break Generation and Detection.
- Internal Diagnostic Capabilities.
 - Loopback Controls for Communications Link Fault Isolation.
 - Break, Parity, Overrun, Framing Error Simulation.
- Full Prioritized Interrupt System Controls.

All communications protocol is a function of the system micro-code and must be loaded before the adapter is operational. All pacing of the interface and control signal status must be handled by the system software.

Asynchronous Communications Block Diagram

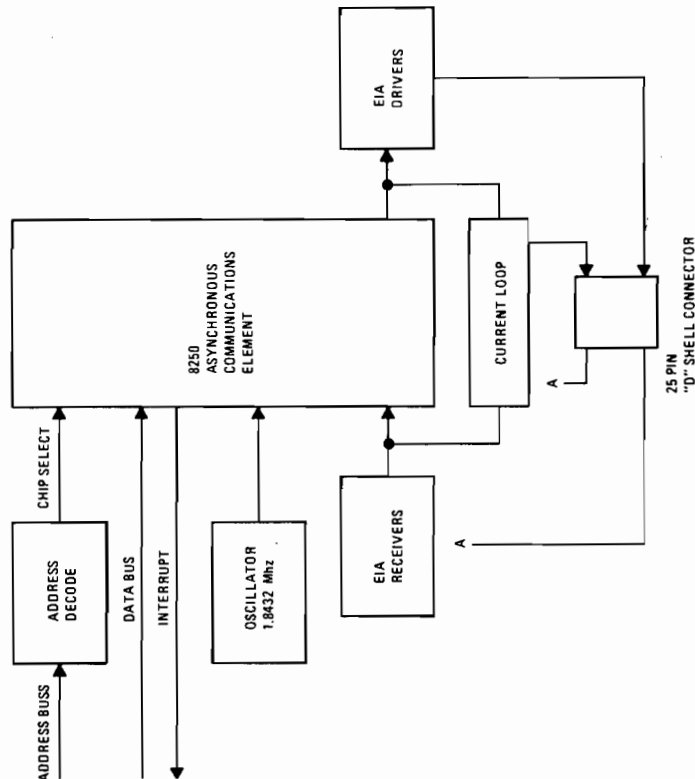


Figure 22. ASYNCHRONOUS COMMUNICATIONS ADAPTER BLOCK DIAGRAM

Modes of Operation

The different modes of operation are selected by programming the 8250 Asynchronous Communications Element. This is done by selecting the I/O address (3F8 to 3FF) and writing data out to the card. Address bit A0, A1 and A2 select the different registers which define the modes of operation. Also, the Divisor Latch Access Bit (Bit 7) of the line control register is used to select certain registers.

I/O Decode for Communications Adapter

Table 21. I/O Decodes (3F8 • 3FF)

| I/O DECODE | REGISTER SELECTED | DLAB STATE |
|------------|------------------------------------|----------------|
| 3F8 | TX BUFFER | DLAB=0 (WRITE) |
| 3F8 | RX BUFFER | DLAB=0 (READ) |
| 3F8 | DIVISOR LATCH LSB | DLAB=1 |
| 3F9 | DIVISOR LATCH MSB | DLAB=1 |
| 3F9 | INTERRUPT ENABLE REGISTER | DLAB=0 |
| 3FA | INTERRUPT IDENTIFICATION REGISTERS | |
| 3FB | LINE CONTROL REGISTER | |
| 3FC | MODEM CONTROL REGISTER | |
| 3FD | LINE STATUS REGISTER | |
| 3FE | MODEM STATUS REGISTER | |

ADDRESS BITS

| A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | DLAB | REGISTER |
|----|----|----|----|----|----|----|----|----|----|------|--|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | Receive Buffer (read), Transmit Holding Reg. (write) |
| | | | | | | | 0 | 0 | 1 | 0 | Interrupt Enable |
| | | | | | | | 0 | 1 | 0 | X | Interrupt Identification |
| | | | | | | | 0 | 1 | 1 | X | Line Control |
| | | | | | | | 1 | 0 | 0 | X | Modem Control |
| | | | | | | | 1 | 0 | 1 | X | Line Status |
| | | | | | | | 1 | 1 | 0 | X | Modem Status |
| | | | | | | | 1 | 1 | 1 | 1 | None |
| | | | | | | | 0 | 0 | 0 | 1 | Divisor Latch (LSB) |
| | | | | | | | 0 | 0 | 1 | 1 | Divisor Latch (MSB) |

A2, A1 and A0 bits are "Don't Cares" and are used to select the different register of the communications chip.

Interrupts

One interrupt line is provided to the system. This interrupt is IRQ4 and will be positive active. To allow the communications card to send interrupts to the system, Bit 3 of the Modem Control Register must be set = 1 (high). At this point, any interrupts allowed by the Interrupt Enable Register will cause an interrupt.

The data format will be as follows:

TRANSMITTER OUTPUT AND RECEIVER INPUT



Data Bit 0 is the first bit to be transmitted or received. The adapter automatically inserts the start bit, the correct parity bit if programmed to do so, and the stop bit (1, 1-1/2 or 2 depending on the command in the Line Control Register).

Interface Description

The communications adapter provides an EIA RS-232-C, or equivalent, interface. One 25 pin "D" shell, male type connector is provided to attach various peripheral devices. In addition, a current loop interface is also located in this same connector. A jumper block is provided to manually select either the voltage interface, or the current loop interface.

- Pin 18 + receive current loop data (20Ma)
- Pin 25 - receive current loop return (20Ma)
- Pin 9 + transmit current loop return (20Ma)
- Pin 11 - transmit current loop data (20Ma)

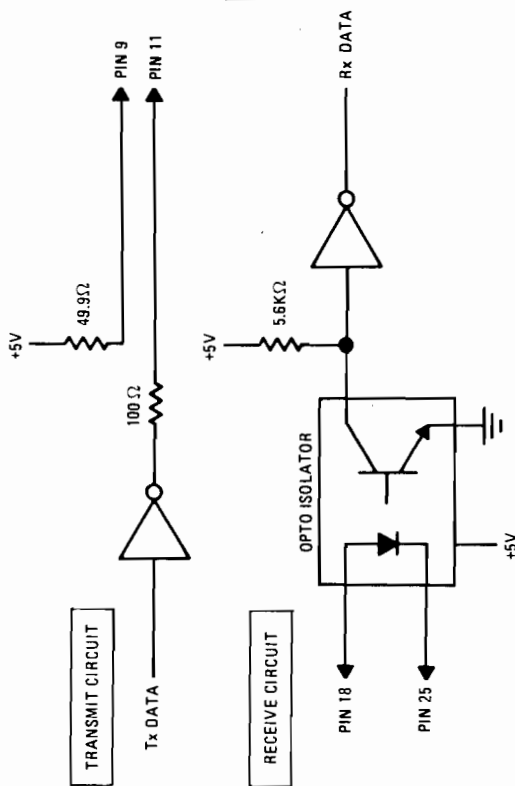


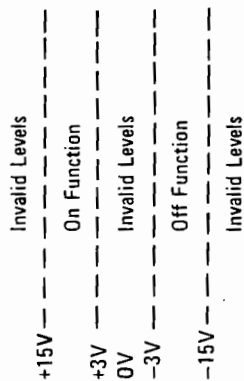
Figure 23. CURRENT LOOP INTERFACE

The voltage interface is a serial interface. It supports certain data and control signals as listed below:

- Pin 2 Transmit Data
- Pin 3 Receive Data
- Pin 4 Request to Send
- Pin 5 Clear to Send
- Pin 6 Data Set Ready
- Pin 7 Signal Ground
- Pin 8 Carrier Detect
- Pin 20 Data Terminal Ready
- Pin 22 Ring Indicate

The adapter converts these signals from TTL to EIA voltage levels and from EIA to TTL voltage levels. These signals are sampled or generated by the communication control chip. These signals can then be sensed by the system software to determine the state of the interface or peripheral device.

| Interchange Voltage | Binary State | Signal Condition | Interface Control Function |
|---------------------|--------------|------------------|----------------------------|
| Positive Voltage = | Binary (0) | = Spacing | =On |
| Negative Voltage = | Binary (1) | = Marking | =Off |



The signal will be considered in the "marking" condition when the voltage on the interchange circuit, measured at the interface point, is more negative than minus three volts with respect to signal ground. The signal will be considered in the "spacing" condition when the voltage is more positive than plus three volts with respect to signal ground. The region between plus three volts and minus three volts is defined as the transition region and is considered an invalid level. The voltage which is more negative than -15V or more positive than +15V will be considered in invalid levels.

During the transmission of data, the "marking" condition will be used to denote the binary state "one" and "spacing" condition will be used to denote the binary state "zero".

For interface control circuits, the function is "on" when the voltage is more positive than +3V with respect to signal ground and is "off" when the voltage is more negative than -3V with respect to signal ground.

INS8250 Functional Pin Description

The following describes the function of all INS8250 input/output pins. Some of these descriptions reference internal circuits.

Note: In the following descriptions, a low represents a logic 0 (0 volt nominal) and a high represents a logic 1 (+2.4 volts nominal).

Input Signals

Chip Select (CS0, CS1, CS2), Pins 12-14: When CS0 and CS1 are high and CS2 is low, the chip is selected. Chip selection is complete when the decoded chip select signal is latched with an active (low) Address Strobe (ADS) input. This enable communication between the INS8250 and the CPU.

Data Input Strobe (DISTR, $\overline{\text{DISTR}}$) Pins 22 and 21: When DISTR is high or $\overline{\text{DISTR}}$ is low while the chip is selected, allows the CPU to read status information or data from a selected register of the INS8250.

Note: Only an active DISTR or $\overline{\text{DISTR}}$ input is required to transfer data from the INS8250 during a read operation. Therefore, tie either the DISTR input permanently low or the DISTR input permanently high, if not used.

Data Output Strobe (DOSTR, $\overline{\text{DOSTR}}$), Pins 19 and 18: When DOSTR is high or $\overline{\text{DOSTR}}$ is low while the chip is selected, allows the CPU to write data or control words into a selected register of the INS8250.

Note: Only an active DOSTR or $\overline{\text{DOSTR}}$ input is required to transfer data to the INS8250 during a write operation. Therefore, tie either the DOSTR input permanently low or the DOSTR input permanently high, if not used.

Address Strobe (ADS), Pin 25: When low, provides latching for the Register Select (A0, A1, A2) and Chip Select (CS0, CS1, CS2) signals.

Note: An active ADS input is required when the Register Select (A0, A1, A2) signals are not stable for the duration of a read or write operation. If not required, tie the ADS input permanently low.

Register Select (A0, A1, A2), Pins 26-28: These three inputs are used during a read or write operation to select an INS8250 register to read from or write into as indicated in the table below. Note that the state of the Divisor Latch Access Bit (DLAB), which is the most significant bit of the Line Control Register, affects the selection of certain INS8250 registers. The DLAB must be set high by the system software to access the Baud Generator Divisor Latches.

| DLAB | A2 | A1 | A0 | Register |
|------|----|----|----|--|
| 0 | 0 | 0 | 0 | Receiver Buffer (read), Transmitter Holding Register (write) |
| 0 | 0 | 0 | 1 | Interrupt Enable |
| X | 0 | 1 | 0 | Interrupt Identification (read only) |
| X | 0 | 1 | 1 | Line Control |
| X | 1 | 0 | 0 | MODEM Control |
| X | 1 | 0 | 1 | Line Status |
| X | 1 | 1 | 0 | MODEM Status |
| X | 1 | 1 | 1 | None |
| 1 | 0 | 0 | 0 | Divisor Latch (least significant byte) |
| 1 | 0 | 0 | 1 | Divisor Latch (most significant byte) |

Master Reset (MR), Pin 35: When high, clears all the registers (except the Receiver Buffer, Transmitter Holding, and Divisor Latches), and the control logic of the INS8250. Also, the state of various output signals (SOUT, INTRPT, OUT 1, OUT 2, RTS, DTR) are affected by an active MR input. (Refer to Table 1.)

Receiver Clock (RCLK), Pin 9: This input is the 16x baud rate clock for the receiver section of the chip.

Serial Input (SIN), Pin 10: Serial data input from the communications link (peripheral device, MODEM, or data set).

Clear to Send (CTS), Pin 36: The CTS signal is a MODEM control function input whose condition can be tested by the CPU by reading Bit 4 (CTS) of the MODEM Status Register. Bit 0 (DCTS) of the MODEM Status Register indicates whether the CTS input has changed state since the previous reading of the MODEM Status Register.

Note: Whenever the CTS bit of the MODEM Status Register changes state, an interrupt is generated if the MODEM Status Interrupt is enabled.

Data Set Ready (DSR), Pin 37: When low, indicates that the MODEM or data set is ready to establish the communications link and transfer data with the INS8250. The DSR signal is a MODEM-control function input whose condition can be tested by the CPU by reading Bit 5 (DSR) of the MODEM Status Register. Bit 1 (DDSR) of the MODEM Status Register indicates whether the DSR input has changed state since the previous reading of the MODEM Status Register.

Note: Whenever the DSR bit of the MODEM Status Register changes state, an interrupt is generated if the MODEM Status Interrupt is enabled.

Received Line Signal Detect (RLSD), Pin 38: When low, indicates that the data carrier has been detected by the MODEM or data set. The RLSD signal is a MODEM-Control function input whose condition can be tested by the CPU by reading Bit 7 (RLSD) of the MODEM Status Register. Bit 3 (DRLSD) of the MODEM Status Register indicates whether the RLSD input has changed state since the previous reading of the MODEM Status Register.

Note: Whenever the RLSD bit of the MODEM Status Register changes state, an interrupt is generated if the MODEM Status Interrupt is enabled.

Ring Indicator (RI), Pin 39: When low, indicates that a telephone ringing signal has been received by the MODEM or data set. The RI signal is a MODEM-control function input whose condition can be tested by the CPU by reading Bit 6 (RI) of the MODEM Status Register. Bit 2 (TERI) of the MODEM Status Register indicates whether the RI input has changed from a low to a high state since the previous reading of the MODEM Status Register.

Note: Whenever the RI bit of the MODEM Status Register changes from a high to a low state, an interrupt is generated if the MODEM Status Interrupt is enabled.

VCC, Pin 40: +5 volt supply.

VSS, Pin 20: Ground (0-volt) reference.

Output Signals

Data Terminal Ready (DTR), Pin 33: When low, informs the MODEM or data set that the INS8250 is ready to communicate. The DTR output signal can be set to an active low by programming Bit 0 (DTR) of the MODEM Control Register to a high level. The DTR signal is set high upon a Master Reset operation.

Request to Send (RTS), Pin 32: When low, informs the MODEM or data set that the INS8250 is ready to transmit data. The RTS output signal can be set to an active low by programming Bit 1 (RTS) of the MODEM Control Register. The RTS signal is set high upon a Master Reset operation.

Output 1 (OUT 1), Pin 34: User-designated output that can be set to an active low by programming Bit 2 (OUT 1) of the MODEM Control Register to a high level. The OUT 1 signal is set high upon a Master Reset operation.

Output 2 (OUT 2), Pin 31: User-designated output that can be set to an active low by programming Bit 3 (OUT 2) of the MODEM Control Register to a high level. The OUT 2 signal is set high upon a Master Reset operation.

Chip Select Out (CSOUT), Pin 24: When high, indicates that the chip has been selected by active CS0, CS1, and CS2 inputs. No data transfer can be initiated until the CSOUT signal is a logic 1.

Driver Disable (DDIS), Pin 23: Goes low whenever the CPU is reading data from the INS8250. A high-level DDIS output can be used to disable an external transceiver (if used between the CPU and INS8250 on the D7-D0 Data Bus) at all times, except when the CPU is reading data.

Baud Out (BAUDOUT), Pin 15: 16x clock signal for the transmitter section of the INS8250. The clock rate is equal to the main reference oscillator frequency divided by the specified divisor in the Baud Generator Divisor Latches. The BAUDOUT may also be used for the receiver section by typing this output to the RCLK input of the chip.

Interrupt (INTRPT), Pin 30: Goes high whenever any one of the following interrupt types has an active high condition and is enabled via the IER: Receiver Error Flag; Received Data Available; Transmitter Holding Register Empty; and MODEM Status. The INTRPT Signal is reset low upon the appropriate interrupt service or a Master Reset operation.

Serial Output (SOUT), Pin 11: Composite serial data output to the communications link (peripheral, MODEM or data set). The SOUT signal is set to the Marking (Logic 1) state upon a Master Reset operation.

Input/Output Signals

Data (D7-D0) Bus, Pins 1-8: This bus comprises eight TRI-STATE input/output lines. The bus provides bidirectional communications between the INS8250 and the CPU. Data, control words, and status information are transferred via the D7-D0 Data Bus.

External Clock Input/Output (XTAL1, XTAL2, Pins 16 and 17: These two pins connect the main timing reference (crystal or signal clock) to the INS8250.

Programming Considerations

Table 22. Asynchronous Communications Reset Functions

| Register/Signal | Reset Control | Reset State |
|-----------------------------------|-----------------------|---|
| Interrupt Enable Register | Master Reset | All Bits Low (0-3 Forced and 4-7 Permanent) |
| Interrupt Identification Register | Master Reset | Bit 0 is High, Bits 1 and 2 Low Bits 3-7 are Permanently Low |
| Line Control Register | Master Reset | All Bits Low |
| MODEM Control Register | Master Reset | All Bits Low |
| Line Status Register | Master Reset | Except Bits 5 & 6 are High |
| MODEM Status Register | Master Reset | Bits 0-3 Low Bits 4-7 - Input Signal |
| SOUT | Master Reset | High |
| INTRPT (RCVR Errs) | Read LSR/MR | Low |
| INTRPT (RCVR Data Ready) | Read RBR/MR | Low |
| INTRPT (RCVR Data Ready) | Read IIR/Write THR/MR | Low |
| INTRPT (MODEM Status Changes) | Read MSR/MR | Low |
| OUT 2 | Master Reset | High |
| RTS | Master Reset | High |
| DTR | Master Reset | High |
| OUT 1 | Master Reset | High |

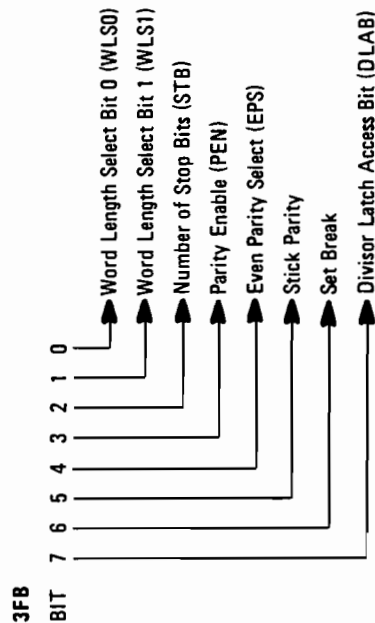
INS8250 Accessible Registers

The system programmer may access or control any of the INS8250 registers via the CPU. These registers are used to control INS8250 operations and to transmit and receive data.

INS8250 Line Control Register

The system programmer specifies the format of the asynchronous data communications exchange via the Line Control Register. In addition to controlling the format, the programmer may retrieve the contents of the Line Control Register for inspection. This feature simplifies system programming and eliminates the need for separate storage in system memory of the line characteristics. The contents of the Line Control Register are indicated and described below.

Line Control Register (LCR)



Bit 0 and 1: These two bits specify the number of bits in each transmitted or received serial character. The encoding of bits 0 and 1 is as follows:

| Bit 1 | Bit 0 | Word Length |
|-------|-------|-------------|
| 0 | 0 | 5 Bits |
| 0 | 1 | 6 Bits |
| 1 | 0 | 7 Bits |
| 1 | 1 | 8 Bits |

Bit 2: This bit specifies the number of Stop bits in each transmitted or received serial character. If bit 2 is a logic 0, 1 Stop bit is generated or checked in the transmit or receive data, respectively. If bit 2 is logic 1 when a 5-bit word length is selected via bits 0 and 1, 1-1/2 Stop bits are generated or checked. If bit 2 is logic 1 when either a 6-, 7-, or 8-bit word length is selected, 2 Stop bits are generated or checked.

Bit 3: This bit is the Parity Enable bit. When bit 3 is a logic 1, a Parity bit is generated (transmit data) or checked (receive data) between the last data word bit and Stop bit of the serial data. (The Parity bit is used to produce an even or odd number of 1's when the data word bits and the Parity bit are summed.)

Bit 4: This bit is the Even Parity Select bit. When bit 3 is a logic 1 and bit 4 is a logic 0, an odd number of logic 1's is transmitted or checked in the data word bits and Parity bit. When bit 3 is a logic 1 and bit 4 is a logic 1, an even number of bits is transmitted or checked.

Bit 5: This bit is the Stick Parity bit. When bit 3 is a logic 1 and bit 5 is a logic 1, the Parity bit is transmitted and then detected by the receiver as a logic 0 if bit 4 is a logic 1 or as a logic 1 if bit 4 is a logic 0.

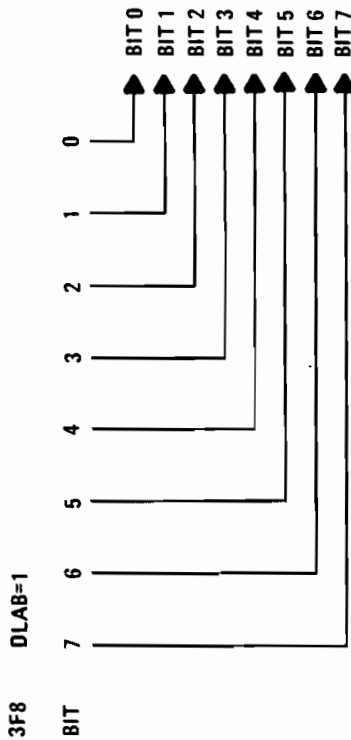
Bit 6: This bit is the Set Break Control bit. When bit 6 is a logic 1, the serial output (SOUT) is forced to the Spacing (logic 0) state and remains there regardless of other transmitter activity. The set break is disabled by setting bit 6 to a logic 0. This feature enables the CPU to alert a terminal in a computer communications system.

Bit 7: This bit is the Divisor Latch Access Bit (DLAB). It must be set high (logic 1) to access the Divisor Latches of the Baud Rate Generator during a Read or Write operation. It must be set low (logic 0) to access the Receiver Buffer, the Transmitter Holding Register, or the Interrupt Enable Register.

INS8250 Programmable Baud Rate Generator

The INS8250 contains a programmable Baud Rate Generator that is capable of taking the clock input (1.8432 Mhz) and dividing it by any divisor from 1 to $(2^{16}-1)$. The output frequency of the Baud Generator is $16 \times \text{the Baud rate} [\text{divisor} \# = (\text{frequency input}) / (\text{baud rate} \times 16)]$. Two 8-bit latches store the divisor in a 16-bit binary format. These Divisor Latches must be loaded during initialization in order to ensure desired operation of the Baud Rate Generator. Upon loading either of the Divisor Latches, a 16-bit Baud counter is immediately loaded. This prevents long counts on initial load.

Divisor Latch Least Significant Bit (DLL)



Divisor Latch Most Significant Bit (DLM)

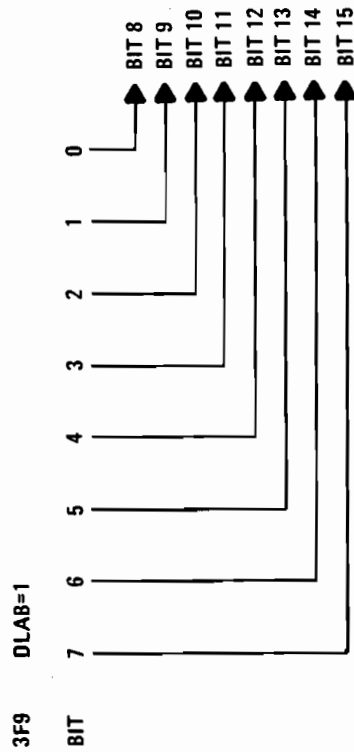


Table 23 illustrates the use of the Baud Rate Generator with a frequency of 1.8432 Mhz. For baud rates of 9600 and below, the error obtained is minimal.

Note: The maximum operating frequency of the Baud Generator is 3.1 Mhz. In no case should the data rate be greater than 9600 Baud.

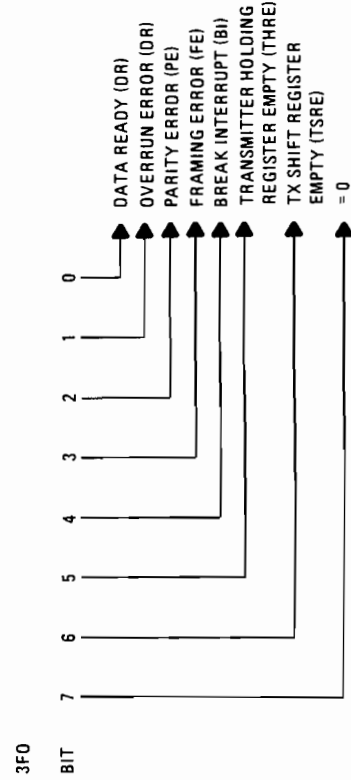
Table 23. BAUD Rate at 1.843 Mhz

| Desired Baud Rate | Divisor Used to Generate 16x Clock | Percent Error Difference Between Desired & Actual |
|-------------------|------------------------------------|---|
| 50 | Decimal 2304 Hex '900' | --- |
| 75 | 1536 '600' | --- |
| 110 | 1047 '417' | --- |
| 134.5 | 857 '359' | 0.026 |
| 150 | 768 '300' | 0.058 |
| 300 | 384 '180' | --- |
| 600 | 192 '0C0' | --- |
| 1200 | 96 '060' | --- |
| 1800 | 64 '040' | --- |
| 2000 | 58 '03A' | 0.69 |
| 2400 | 48 '030' | --- |
| 3600 | 32 '020' | --- |
| 4800 | 24 '018' | --- |
| 7200 | 16 '010' | --- |
| 9600 | 12 '00C' | --- |

Line Status Register

This 8-bit register provides status information to the CPU concerning the data transfer. The contents of the Line Status Register are indicated and described below.

Line Status Register (LSR)



Bit 0: This bit is the receiver Data Ready (DR) indicator. Bit 0 is set to a logic 1 whenever a complete incoming character has been received and transferred into the Receiver Buffer Register. Bit 0 may be reset to a logic 0 either by the CPU reading the data in the Receiver Buffer Register or by writing a logic 0 into it from the CPU.

Bit 1: This bit is the Overrun Error (OE) indicator. Bit 1 indicates that data in the Receiver Buffer Register was not read by the CPU before the next character was transferred into the Receiver Buffer Register, thereby destroying the previous character. The OE indicator is reset whenever the CPU reads the contents of the Line Status Register.

Bit 2: This bit is the Parity Error (PE) indicator. Bit 2 indicates that the received data character does not have the correct even or odd parity, as selected by the even parity-select bit. The PE bit is set to a logic 1 upon detection of a parity error and is reset to a logic 0 whenever the CPU reads the contents of the Line Status Register.

Bit 3: This bit is the Framing Error (FE) indicator. Bit 3 indicates that the received character did not have a valid Stop bit. Bit 3 is set to a logic 1 whenever the Stop bit following the last data bit or parity bit is detected as a zero bit (Spacing level).

Bit 4: This bit is the Break Interrupt (BI) indicator. Bit 4 is set to a logic 1 whenever the received data input is held in the Spacing (logic 0) state for longer than a full word transmission time (that is, the total time of Start bit + data bits + Parity + Stop bits).

Note: Bits 1 through 4 are the error conditions that produce a Receiver Line Status interrupt whenever any of the corresponding conditions are detected.

Bit 5: This bit is the Transmitter Holding Register Empty (THRE) indicator. Bit 5 indicates that the INS8250 is ready to accept a new character for transmission. In addition, this bit causes the INS8250 to issue an interrupt to the CPU when the Transmitter Holding Register Empty Interrupt enable is set high. The THRE bit is set to a logic 1 when a character is transferred from the Transmitter Holding Register into the Transmitter Shift Register. The bit is reset to logic 0 concurrently with the loading of the Transmitter Holding Register by the CPU.

Bit 6: This bit is the Transmitter Shift Register Empty (TSRE) indicator. Bit 6 is set to a logic 1 whenever the Transmitter Shift Register is idle. It is reset to logic 0 upon a data transfer from the Transmitter Holding Register to the Transmitter Shift Register. Bit 6 is a read-only bit.

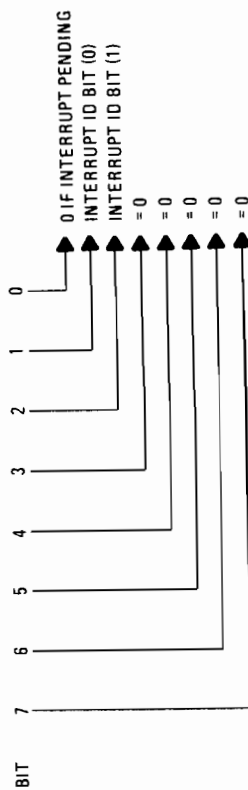
Interrupt Identification Register

The INS8250 has an on-chip interrupt capability that allows for complete flexibility in interfacing to all the popular microprocessors presently available. In order to provide minimum software overhead during data character transfers, the INS8250 prioritizes interrupts into four levels. The four levels of interrupt conditions are as follows: Receiver Line Status (priority 1); Received Data Ready (priority 2); Transmitter Holding Register Empty (priority 3); and MODEM Status (priority 4).

Information indicating that a prioritized interrupt is pending and the type of that interrupt are stored in the Interrupt Identification Register (refer to Table 5). The Interrupt Identification Register (IIR), when addressed during chip-select time, freezes the highest priority interrupt pending and no other interrupts are acknowledged until that particular interrupt is serviced by the CPU. The contents of the IIR are indicated and described below.

Interrupt Identification Register (IIR)

3FA



Bit 0: This bit can be used in either a hardwired prioritized or polled environment to indicate whether an interrupt is pending. When bit 0 is a logic 0, an interrupt is pending and the IIR contents may be used as a pointer to the appropriate interrupt service routine. When bit 0 is a logic 1, no interrupt is pending and polling (if used) continued.

Bits 1 and 2: These two bits of the IIR are used to identify the highest priority interrupt pending as indicated in Table 5.

Bits 3 through 7: These five bits of the IIR are always logic 0.

Table 24. Interrupt Control Functions (Asynchronous)

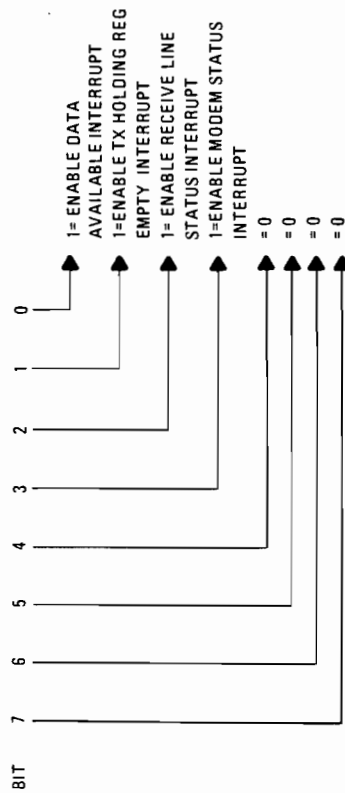
| Interrupt ID Register | | | | Interrupt Set and Reset Functions | | |
|-----------------------|-------|-------|----------------|------------------------------------|--|--|
| Bit 2 | Bit 1 | Bit 0 | Priority Level | Interrupt Type | Interrupt Source | Interrupt Reset Control |
| 0 | 0 | 1 | — | None | None | — |
| 1 | 1 | 0 | Highest | Receiver Line Status | Overrun Error or Parity Error or Framing Error or Break Interrupt | Reading the Line Status Register |
| 1 | 0 | 0 | Second | Received Data Available | Receiver Data Available | Reading the Receiver Buffer Register |
| 0 | 1 | 0 | Third | Transmitter Holding Register Empty | Transmitter Holding Register Empty | Reading the IIR Register (if source of interrupt) or Writing into the Transmitter Holding Register |
| 0 | 0 | 0 | Fourth | MODEM Status | Clear to Send or Data Set Ready or Ring Indicator or Received Line Signal Detect | Reading the MODEM Status Register |

Interrupt Enable Register

This 8-bit register enables the four types of interrupt of the INS8250 to separately activate the chip Interrupt (INTRPT) output signal. It is possible to totally disable the interrupt system by resetting bits 0 through 3 of the Interrupt Enable Register. Similarly, by setting the appropriate bits of this register to a logic 1, selected interrupts can be enabled. Disabling the interrupt system inhibits the Interrupt Identification Register and the active (high) INTRPT output from the chip. All other system functions operate in their normal manner, including the setting of the Line Status and MODEM Status Registers. The contents of the Interrupt Enable Register are indicated and described below.

Interrupt Enable Register (IER)

3F9 DLAB=0



Bit 0: This bit enables the Received Data Available Interrupt when set to logic 1.

Bit 1: This bit enables the Transmitter Holding Register Empty Interrupt when set to logic 1.

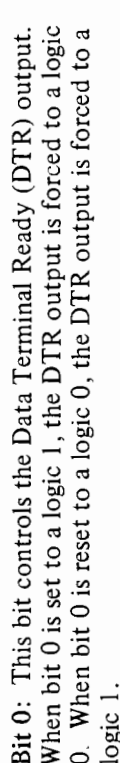
Bit 2: This bit enables the Receiver Line Status Interrupt when set to logic 1.

Bit 3: This bit enables the MODEM Status Interrupt when set to logic 1.

Bits 4 through 7: These four bits are always logic 0.

In the diagnostic mode, the receiver and transmitter interrupts are fully operational. The MODEM Control Interrupts are also operational but the interrupts' sources are now the lower four bits of the MODEM Control Register instead of the four MODEM Control inputs. The interrupts are still controlled by the Interrupt Enable Register.

MODEM Control Register (MCR)



Bit 1: This bit controls the Request to Send (RTS) output. Bit 1 affects the RTS output in a manner identical to that described above for bit 0.

Bit 3: This bit controls the Output 2 (OUT 2) signal, which is an auxiliary user-designated output. Bit 3 affects the OUT 2 output in a manner identical to that described above for bit 0.

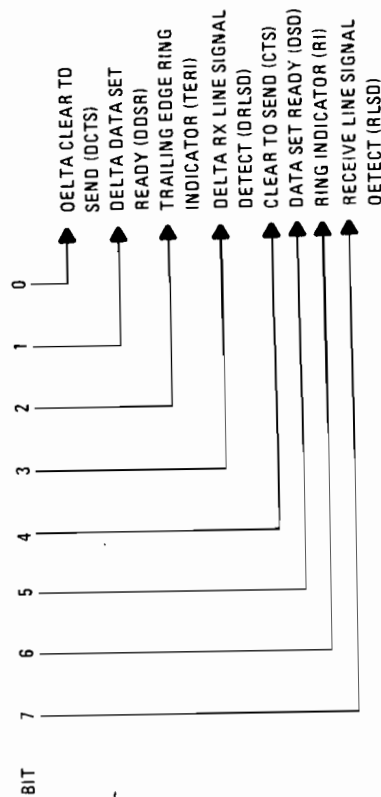
2-160

The INS8250 interrupt system can be tested by writing into the lower four bits of the MODEM Status Register. Setting any of these bits to a logic 1 generates the appropriate interrupt (if enabled). The resetting of these interrupts is the same as in normal INS8250 operation. To return to normal operation, the registers must be reprogrammed for normal operation, and then bit 4 of the MODEM Control Register must be reset to logic 0.

MODEM Status Register

The content of the MODEM Status Register are indicated and described below.

3FE



Bit 0: This bit is the Delta Clear to Send (DCTS) indicator. Bit 0 indicates that the CTS input to the chip has changed state since the last time it was read by the CPU.

Bit 1: This bit is the Delta Data Set Ready (DDSR) indicator.

Bit 1 indicates that the DSR input to the chip has changed state since the last time it was read by the CPU.

Bit 2: This bit is the Trailing Edge of Ring Indicator (TERI) detector. Bit 2 indicates that the RI input to the chip has changed from an On (logic 1) to an Off (logic 0) condition.

Bit 3: This bit is the Delta Received Line Signal Detector (DRLSD) indicator. Bit 3 indicates that the RLSD input to the chip has changed state.

Note: Whenever bit 0, 1, 2, or 3 is set to a logic 1, a MODEM Status interrupt is generated.

Bit 4: This bit is the complement of the Clear to Send (CTS) input. If bit 4 (loop) of the MCR is set to a 1, this bit is equivalent to RTS in the MCR.

Bit 5: This bit is the complement of the Data Set Ready (DSR) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to DTR in the MCR.

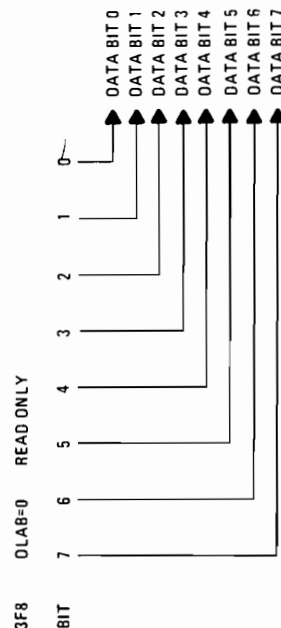
Bit 6: This bit is the complement of the Ring Indicator (RI) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to OUT 1 in the MCR.

Bit 7: This bit is the complement of the Received Line Signal Detect (RLSD) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to OUT 2 of the MCR.

Receiver Buffer Register

The Receiver Buffer Register contains the received character as defined below.

Receiver Buffer Register (RBR)

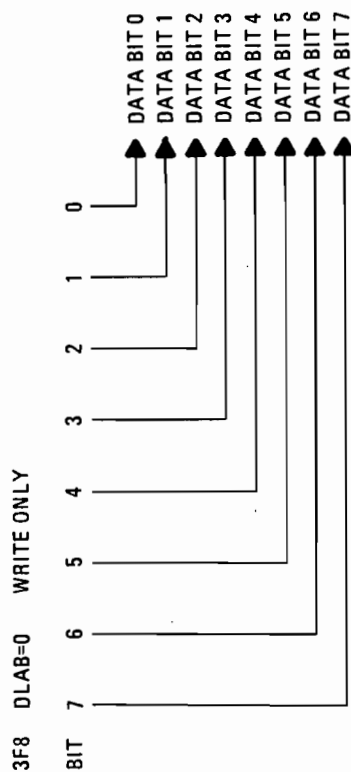


Bit 0 is the least significant bit and is the first bit serially received.

Transmitter Holding Register

The Transmitter Holding Register contains the character to be serially transmitted and is defined below:

Transmitter Holding Register (THR)



Bit 0 is the least significant bit and is the first bit serially transmitted.

Selecting the Interface Format and Adapter Address

The Voltage or Current loop interface and Adapter Address are selected by plugging the programmed shunt modules with the locator dots up or down. See the figure below for the configurations.

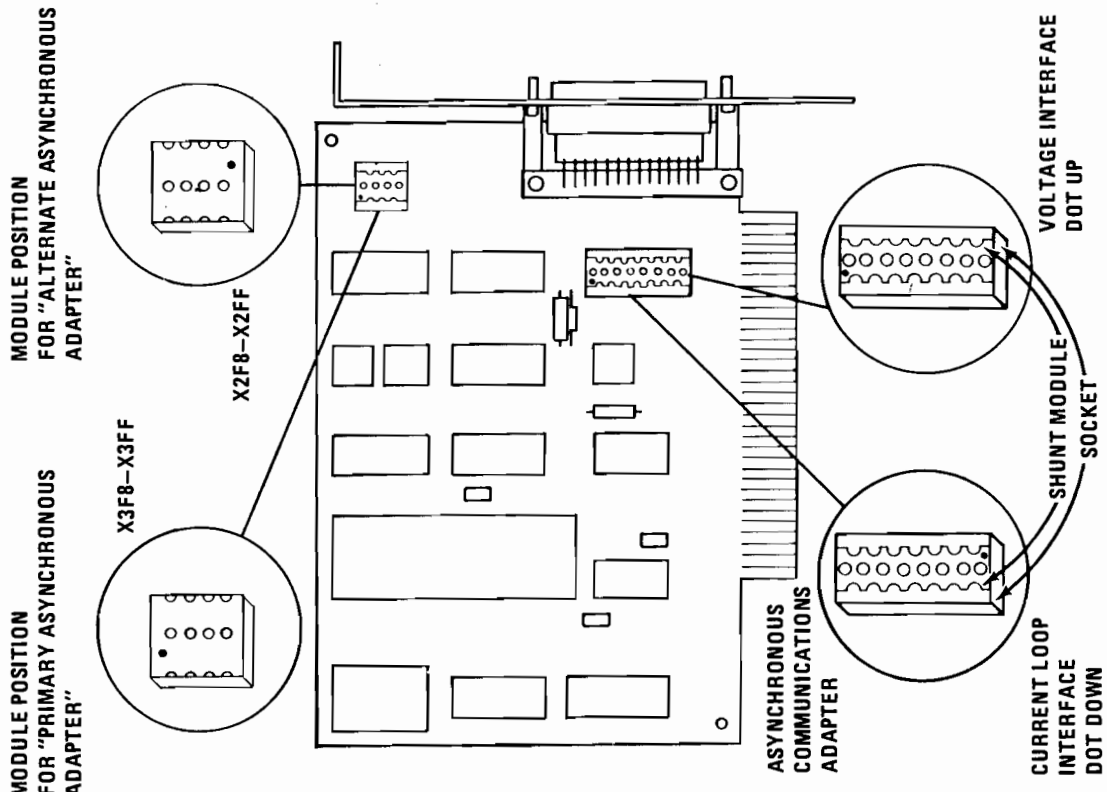
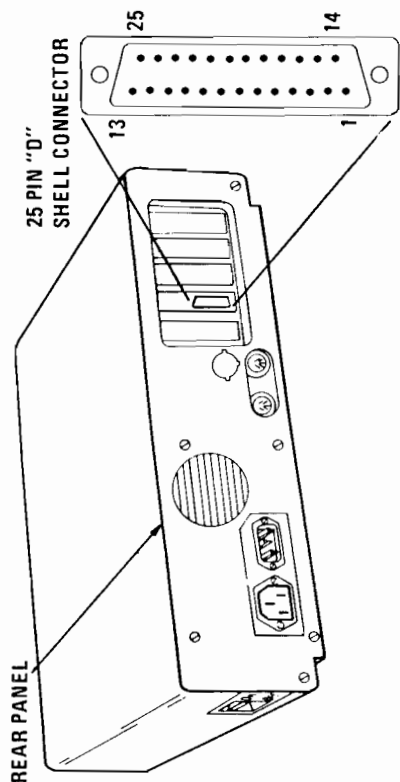


Figure 23. SELECTING THE INTERFACE FORMAT AND ADAPTER ADDRESS

Asynchronous Communications Adapter Connector Interface Specifications



HARDWARE

AT STANDARD TTL LEVELS

| Pin | Description |
|-----|---------------------------------------|
| 1 | NC |
| 2 | Transmit Data |
| 3 | Receive Data |
| 4 | Request to send |
| 5 | Clear to send |
| 6 | Data set ready |
| 7 | Signal ground |
| 8 | Carrier detect |
| 9 | +Transmit current loop return (20 ma) |
| 10 | NC |
| 11 | - Transmit current loop data (20 ma) |
| 12 | NC |
| 13 | NC |
| 14 | NC |
| 15 | NC |
| 16 | NC |
| 17 | NC |
| 18 | +Receive current loop data (20 ma) |
| 19 | NC |
| 20 | Data Terminal Ready |
| 21 | NC |
| 22 | Ring Indicate |
| 23 | NC |
| 24 | NC |
| 25 | - Receive current loop return (20 ma) |

NOTE: To avoid inducing voltage surges on interchange circuits, signals from interchange circuits shall not be used to drive inductive devices, such as relay coils.

Prototype Card

The Prototype Card is 4.2 inches high x 13.2 inches long and plugs into a System Expansion Slot. All system control signals and voltage requirements are provided through a 2 x 31 position card edge tab.

The card contains a voltage bus (+5V) and a ground bus (0V). Each bus borders the card, with the voltage bus on the back (pin side) and the ground bus on the front (component side). A system interface design is also provided on the Prototype Card. The logic diagram for this interface is provided in Appendix D. The Prototype Card can also accommodate a D shell connector if it is needed. The connector size can range from a 9 to a 37 position connector.

Note: Install all components on the component side of the Prototype Card. The total width of the card including components should not exceed 0.700". If these specifications are not met, components on the Prototype Card may touch other cards plugged into adjacent slots.

Prototype Card Block Diagram

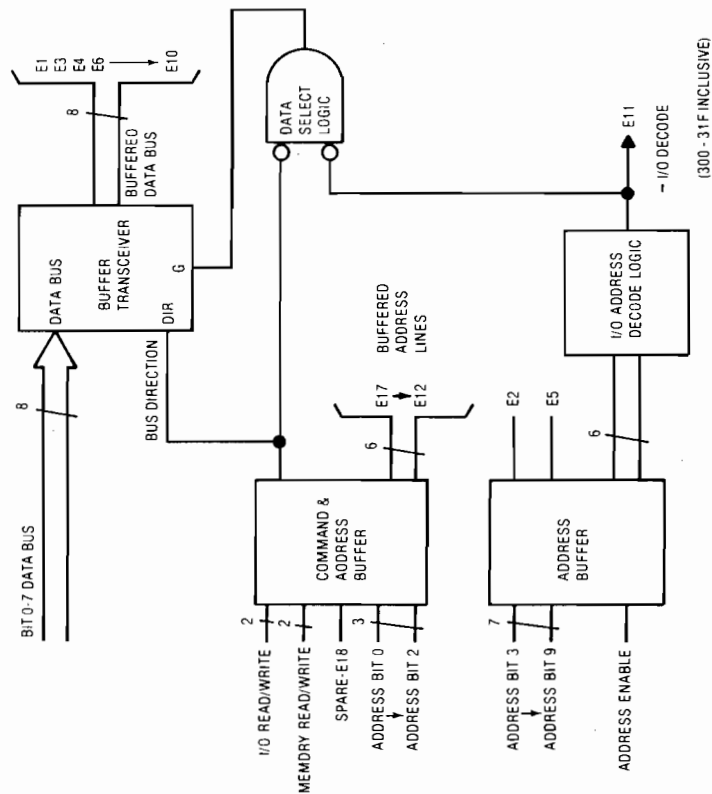


Figure 25. PROTOTYPE CARD BLOCK DIAGRAM

I/O Channel Interface

The Prototype Card has two layers screened onto it (one on the front and one on the back). It also has 3,909 plated through holes that are .040" in size and have a .060" pad which are located on a 0.10" grid. There are 37 plated through holes that are .048" in size. These holes are located at the rear of the card (viewed as if installed in the machine). These 37 holes are used for a 9 to 37 position D shell connector. The card also has 5 holes that are 0.125" in size. One hole is located just above the two rows of D shell connector holes, and the other four are located in the corners of the board (one in each corner).

Prototype Card Layout

The component side has the ground bus (.050" wide) screened on it and card edge tabs that are labeled A1 through A31.

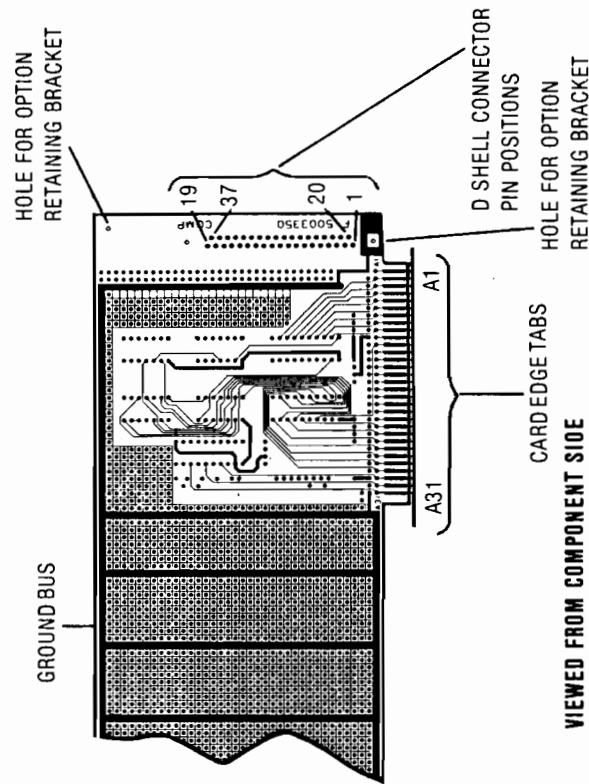
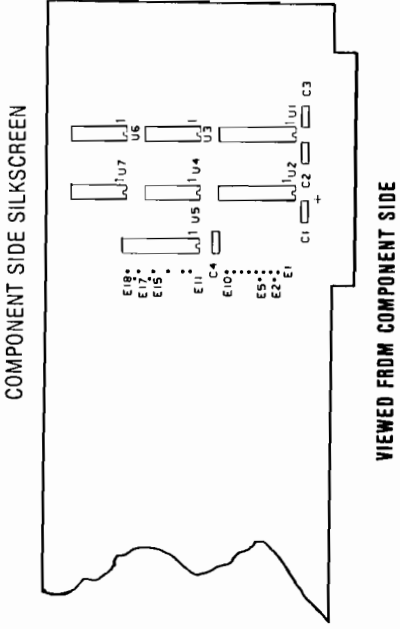


Figure 26. I/O CHANNEL INTERFACE

The component side also has a silk screen printed on it that is used as a component guide for the I/O interface.



The pin side has a voltage bus (+5v and .050" wide) screened onto it and card edge tabs that are labeled B1 through B31.

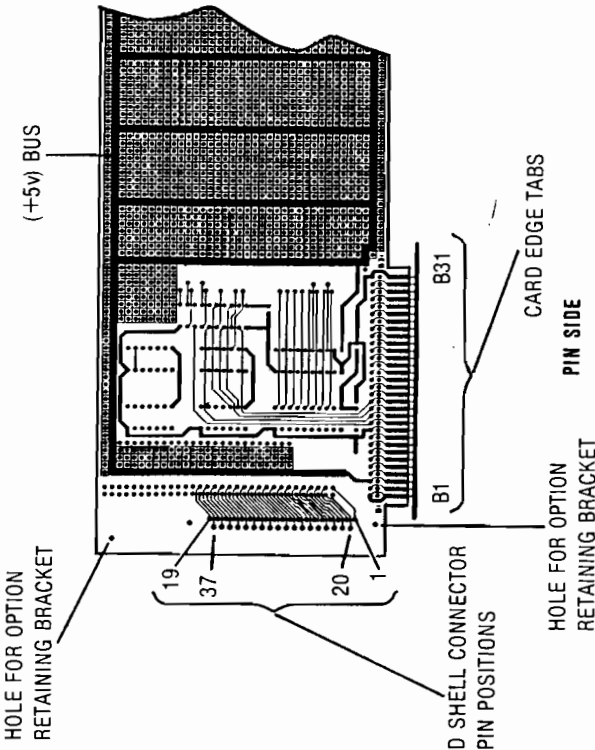


Figure 27. I/O CHANNEL INTERFACE

Each card edge tab is connected to a plated through hole by a .012" land. There are three ground tabs connected to the ground bus by three .012" lands. Also, there are two +5V tabs connected to the voltage bus by two .012" lands.

Prototype Card to System Board Interface

The I/O Channel Description and I/O Channel Diagram are on pages 2-8 through 2-12 of this manual. Also the Prototype Card Interface Logic Diagram is in Appendix D page D-56. If the recommended interface logic is used, the list of TTL type numbers listed below will help you select the necessary components.

| Component | TTL Number | Description |
|-----------|------------|--|
| U1 | 74LS245 | Octal Bus Transceiver |
| U2,U5 | 74LS244 | Octal Buffers Line Driver/Line Receivers |
| U4 | 74LS04 | Hex Inverters |
| U3 | 74LS08 | Quadruple 2 - Input Positive - And Gate |
| U6 | 74LS02 | Quadruple 2 - Input Positive - Nor Gate |
| U7 | 74LS21 | Dual 4 - Input Positive - And Gate |
| C1 | | 10.0 uf Tantalum Capacitor |
| C2,C3,C4 | | .047 uf Ceramic Capacitor |

System Loading and Power Limitations

Because of the number of options that may be installed in the Personal Computer, the I/O bus loading should be limited to one Schottky TTL load. If the interface circuitry on the card is used, then this requirement is met.

The power limitations to be observed are located on pages 2-43 through 2-46 of this manual.

Prototype Card External Interface

If a connector is required for the card function, then you should purchase one of the recommended connectors (manufactured by Amp) or equivalent listed below:

| Connector Size | Part Number (Amp) |
|-------------------------|-------------------|
| 9 pin D shell (Male) | 205865-1 |
| 9 pin D shell (Female) | 205866-1 |
| 15 pin D shell (Male) | 205867-1 |
| 15 pin D shell (Female) | 205868-1 |
| 25 pin D shell (Male) | 205857-1 |
| 25 pin D shell (Female) | 205858-1 |
| 37 pin D shell (Male) | 205859-1 |
| 37 pin D shell (Female) | 205860-1 |

EXAMPLE

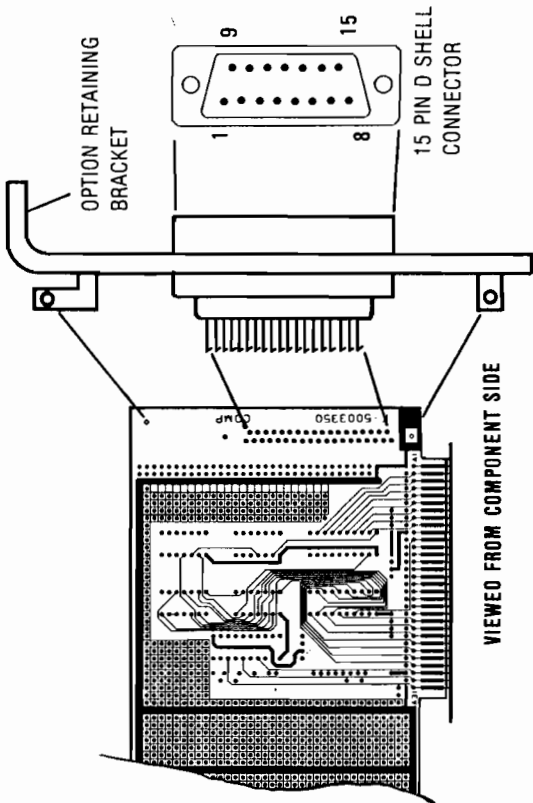


Figure 28. PROTOTYPE CARD EXTERNAL INTERFACE

SECTION 3. ROM and SYSTEM
USAGE

Contents

ROM BIOS 3-2

BIOS Cassette Logic 3-8

Keyboard Encoding and Usage 3-11

Low Memory Maps 3-21

ROM

ROM BIOS

The ROM resident Basic I/O System (BIOS) provides the device level control of the major I/O devices in the System Unit. The BIOS routines allow the assembly language programmer to perform block (diskette and cassette) or character (Video, communications, keyboard and printer) level I/O operations without any concern for device address and operating characteristics. Additionally, system services such as time of day and memory size determination are provided. The goal is to provide an operational interface to the system and relieve the programmer from concern over hardware device characteristics.

Finally the BIOS interface insulates the user from the hardware allowing new devices to be added to the System Unit, yet retaining the BIOS level interface to the device. In this manner, user programs become transparent to hardware modifications and enhancements. A complete listing of the BIOS is provided in Appendix "A".

Use of BIOS

Access to the BIOS function is through the 8088 software interrupts. Each BIOS entry point is available through its own interrupt, which can be found in the interrupt vector listing. The software interrupts 10H through 1AH each access a different BIOS routine. For example, to determine the amount of memory available in the system,

```
INT     12H
```

will invoke the memory size determination routine in BIOS and return the value to the caller.

Parameter Passing

All parameters passed to and from the BIOS routines go through the 8088 registers. The prologue of each BIOS function indicates the registers used on the call and the return. For the memory size example above, no parameters are passed, and the result, memory size in 1K Byte increments is returned in the AX register.

Where a BIOS function has several possible operations, the AH register is used on input to indicate the desired operation. For example, to set the time of day, the following code is required.

```
MOV     AH, 1           ;function is to set time of
                        ;day.
MOV     CX,HIGH_COUNT  ;establish the current time.
MOV     DX,LOW_COUNT
INT     1AH
While to read the time of day:
MOV     AH,0           ;function is to read the time
                        ;of day.
INT     1AH           ;read the timer.
```

As a general rule, the BIOS routines preserve all registers except for AX and the flags. Other registers are modified on return only if they are returning a value to the caller. The exact register usage can be seen in the prologue of each BIOS function.

Interrupt Vector Listing

| Interrupt Number | Name | BIOS Initialization |
|------------------|---------------------------|-----------------------------------|
| 0 | Divide by Zero | None |
| 1 | Single Step | None |
| 2 | Non Maskable | NMI_INT (F000:E2C3) |
| 3 | Breakpoint | None |
| 4 | Overflow | None |
| 5 | Print Screen | PRINT_SCREEN (F000:FF54) |
| 6 | Unused | Unused |
| 7 | Unused | Unused |
| 8 | Time of Day | TIMER_INT (F000:FEA5) |
| 9 | Keyboard | Kb_INT (F000:E987) |
| A | Unused | Unused |
| B | 8259 | Unused |
| C | Interrupt | Unused (Reserved Communications) |
| D | Vectors | Unused |
| E | Unused | Unused |
| F | Unused (Reserved Printer) | DISK_INT (F000:EF57) |
| 10 | Video | VIDEO_I/O (F000:F055) |
| 11 | Equipment Check | EQUIPMENT (F000:F84D) |
| 12 | Memory | MEMORY_SIZE_DETERMINE (F000:F841) |
| 13 | Diskette | DISKETTE_I/O (F000:EC59) |
| 14 | Communications | RS232_I/O (F000:E739) |
| 15 | Cassette | CASSETTE_I/O (F000:F859) |
| 16 | Keyboard | KEYBOARD_I/O (F000:E82E) |
| 17 | Printer | PRINTER_I/O (F000:EF02) |
| 18 | Cassette BASIC | (F600:0000) |
| 19 | Bootstrap | BOOT_STRAP (F000:E6F2) |
| 1A | Time of Day | TIME_OF_DAY (F000:FE6E) |
| 1B | User Supplied | DUMMY_RETURN (F000:FF53) |
| 1C | Routines | DUMMY_RETURN (F000:FF53) |
| 1D | Keyboard Break | Unused |
| 1E | Timer Tick | Unused |
| 1F | Video initialization | VIDEO_PARAMS (F000:F044) |
| | Diskette Parameters | DISK_BASE (F000:EF07) |
| | Video Graphics Chars | None |

Vectors with Special Meanings

Interrupt 1BH – Keyboard Break Address

This vector points to the code to be exercised when the CTRL BREAK keys are depressed on the keyboard. The vector is invoked while responding to the keyboard interrupt, and control should be returned via an IRET instruction. The power on routines initialize this vector to point to an IRET instruction, so that nothing happens when CTRL BREAK keys are depressed unless the application program sets a different value.

Control may be retained by this routine, with the following problems. The BREAK may have occurred during interrupt processing, so that one or more End of Interrupt commands must be set to the 8259 controller. Also, all I/O devices should be reset in case an operation was underway at that time.

Interrupt 1CH – Timer Tick

This vector points to the code to be executed on every tick of the system clock. This vector is invoked while responding to the timer interrupt, and control should be returned via an IRET instruction. The power on routines initialize this vector to point to an IRET instruction, so that nothing happens unless the application modifies the pointer. It is the responsibility of the application to save and restore all registers that will be modified.

Interrupt 1DH – Video Parameters

This vector points to a data region containing the parameters required for the initialization of the 6845 on the video card. Note that there are four separate tables, and all four must be reproduced if all modes of operation are to be supported. The power on routines initialize this vector to point to the parameters contained in the ROM video routine.

Interrupt 1EH – Diskette Parameters

This vector points to a data region containing the parameters required for the diskette drive. The power on routines initialize the vector to point to the parameters contained in the ROM diskette routine. These default parameters represent the specified values for any IBM drives attached to the machine. Changing this parameter block to reflect the specifications of the other drives attached may be necessary.

Interrupt 1FH – Graphics Character Extensions

When operating in the graphics modes of the Color/Graphics Monitor Adapter (320 x 200 or 640 x 200), the read/write character interface will form the character from the ASCII code point, using a set of dot patterns. The dot patterns for the first 128 code points are contained in ROM. To access the other 128 code points, this vector must be established to point at a table of up to 1K bytes, where each code point is represented by 8 bytes of graphic information. At power on this vector is initialized to 0:0, and it is the responsibility of the user to change this vector if the additional code points are required.

Other Read/Write Memory Usage

The IBM ROM BIOS routines use 256 bytes of memory starting at absolute 400 to 4FF. Locations 400-407 contain the base addresses of any RS232 cards attached to the system, 0's if none attached. These locations, in order, represent the 0 to 3 values used as the parameter to the RS232 BIOS routine. Locations 408-40F provide the same function, but for the PRINTER.

Memory locations 300-3FF are used as a stack area during the power on initialization, and the bootstrap, when control passed to it from power on. If the user desires the stack in a different area, it must be set by the application.

Note: Use the Interrupt Vector Listing as an aid to locate these topics in the ROM BIOS listing, Appendix "A".

BIOS Programming Tip

When programming with BIOS you should keep in mind that if an error is reported by the diskette code, to reset the diskette adapter and retry the operation. A specified number of retries should be required on reads to ensure the problem is not due to motor start-up.

BIOS Memory Map

| STARTING ADDRESS HEX | |
|----------------------|-----------------------------|
| 00000 | BIOS INTERRUPT VECTORS |
| 00080 | AVAILABLE INTERRUPT VECTORS |
| 00400 | BIOS DATA AREA |
| 00500 | USER READ/ WRITE MEMORY |
| F4000 | USER READ ONLY MEMORY |
| F6000 | CASSETTE BASIC INTERPRETER |
| FE000 | BIOS PROGRAM AREA |

Figure 29. BIOS MEMORY MAP

BIOS Cassette Logic Software Algorithms

Interrupt 15

The cassette routine will be called with the request type in AH and the address of the bytes to be read or written will be specified by (ES):(BX) and the number of bytes to read/write will be specified by (CX). The actual number of bytes read will be returned in (DX). Read block and write block will automatically turn the motor on at the start and off at the end. The requests are as follows:

- (AH) = 0 Turn the cassette motor on.
- (AH) = 1 Turn the cassette motor off.
- (AH) = 2 (Read Block) Read (CX) bytes into memory beginning at address (ES):(BX) and return actual number of bytes read in (DX). Return the cassette status in (AH).
- (AH) = 3 (Write Block) Write (CX) bytes onto the cassette beginning at address (DS):(BX). Return the cassette status in (AH).

STATUS:

- AH = 00 No errors
- AH = 01 CRC-Error (Read Block)
- AH = 02 No data transitions
- AH = 04 No leader
- AH = 80 Invalid command

Note: The carry flag will be set on any error.

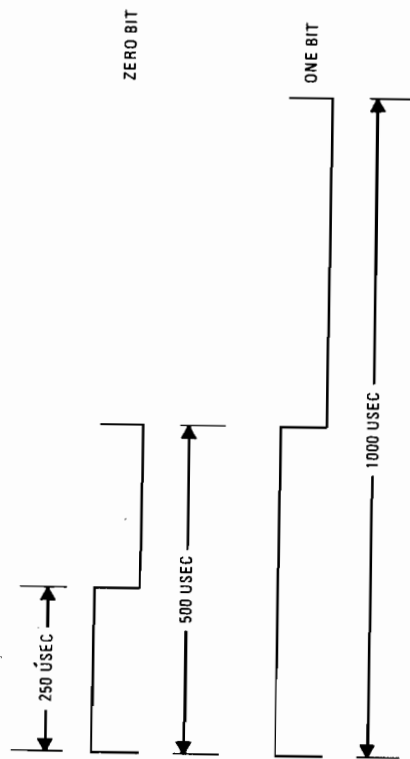
Cassette Write

The WRITE BLOCK routine writes a tape block on the cassette. The tape block is described in Data Record Architecture page (3-10).

The WRITE BLOCK routine turns on the cassette motor and a synchronization bit (0) and then writes 256 bytes of all ones, the leader, to the tape. Next, one or more data blocks are written (depends on number in CX). After each data block of 256 bytes, a two byte CRC is written. The data bytes are taken from the memory location pointed at by ES.

The WRITE BYTE routine disassembles the byte and writes it a bit at a time to the cassette. The method used is to set TIMER 2 to the period of the desired data bit. The timer is set to a period of 1.0 millisecond for a one bit and 0.5 millisecond for a zero bit.

The timer is set mode 3 which means it will output a square wave with period given by its count register. The timer's period is changed on the fly for each data bit to be written to the cassette. If the number of data bytes to be written is not an integral multiple of 256, then after the last desired data byte from memory has been written, the data block will be extended to 256 bytes by writing multiples of the last data byte. The last block will be closed with two CRC bytes as usual. After the last data block, a trailer consisting of four bytes of all one bits will be written. Finally, the motor will be turned off. There are no errors reported by this routine.



Cassette Read

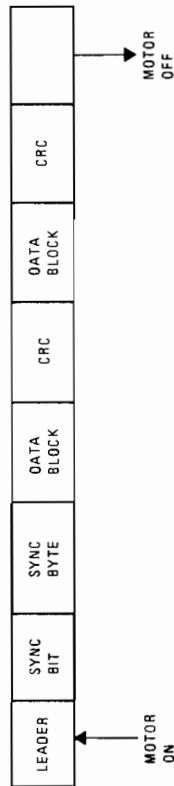
The READ BLOCK routine turns on the cassette motor and then delays for approximately 0.5 secs for it to come up to speed.

The READ BLOCK routine then searches for leader and must detect all one bits for approximately 1/4 of leader length before it can look for the sync byte. If a correct sync byte (X'16') is not found, the routine goes back and searches for leader again. The data is read a bit at a time and assembled into bytes. After each byte is assembled it is written into memory at location ES:BX and then BX is incremented by one.

After each multiple of 256 data bytes are read, the CRC is read and compared to the CRC generated. If a CRC error is detected, the routine will exit with the carry flag set to indicate an error and status (AH)=01 for CRC error. DX will contain the number of bytes written into memory.

Note: The Time of Day Interrupt (IRQ0) is disabled during the cassette read operation.

Data Record Architecture



1. Leader 256 bytes (of ones)
2. Sync byte ASCII Sync Char (X'16')
3. Sync byte (X'16')
4. Data Blocks 256 bytes
5. CRC — 2 bytes — for each data block

Error Recovery

Error recovery is handled by software. A cyclic redundancy check (CRC) is used to detect errors. The polynomial used is:

$$G(X) - X^{16} \ll X^{12} \ll X^5 \ll X^1$$

Which is the polynomial used by the SDLC interface. Essentially, as bits are written/read from tape, they are passed through the CRC-register in software. After a block of data is written, the complemented value of the calculated CRC-register is written on tape. On reading the cassette data, the CRC bytes are read and compared to the generated CRC value. If the read CRC does not equal the generated one, the processor's carry flag is set and status (AH) is set to X'01' to indicate a CRC error has occurred. Also, the routine is exited on CRC error.

Keyboard Encoding and Usage

Encoding

The keyboard routine provided by IBM in ROM BIOS is responsible for converting the keyboard scan codes into what will be termed "Extended ASCII".

Extended ASCII encompasses one byte character codes with possible values of 0-255, an extended code for certain extended keyboard functions and functions that are handled within the keyboard routine or through interrupts.

Character Codes

The following character codes are passed through the BIOS keyboard routine to the system or application program. A "J" means the combination is suppressed in the keyboard routine. The codes are returned in AL. See Appendix C for exact codes. Use keyboard Scan Code diagram for reference page 2-23.

Table 25. Character Codes (U.S. Keyboard Layout)

| KEY # | BASE CASE | UPPER CASE | CTRL | ALT |
|-------|-----------------|-----------------|------------------|--------|
| 1 | ESC | ESC | ESC | -1 |
| 2 | 1 | 1/2 | -1 | Note 1 |
| 3 | 2 | @ | NUL (000) Note 1 | Note 1 |
| 4 | 3 | # | -1 | Note 1 |
| 5 | 4 | \$ | -1 | Note 1 |
| 6 | 5 | % | -1 | Note 1 |
| 7 | 6 | & | RS (030) | Note 1 |
| 8 | 7 | * | -1 | Note 1 |
| 9 | 8 | + | -1 | Note 1 |
| 10 | 9 | (| -1 | Note 1 |
| 11 | 0 |) | -1 | Note 1 |
| 12 | - | - | US (031) | Note 1 |
| 13 | = | + | -1 | Note 1 |
| 14 | Backspace (008) | Backspace (008) | DEL (127) | -1 |
| 15 | Backspace (009) | Backspace (009) | -1 | -1 |
| 16 | q | Q | DC1 (017) | Note 1 |
| 17 | w | W | ETB (023) | Note 1 |
| 18 | e | E | ENQ (005) | Note 1 |
| 19 | r | R | DC2 (018) | Note 1 |
| 20 | t | T | DC4 (020) | Note 1 |
| 21 | y | Y | EM (025) | Note 1 |
| 22 | u | U | NAK (021) | Note 1 |
| 23 | i | I | HT (009) | Note 1 |
| 24 | o | O | SI (015) | Note 1 |
| 25 | p | P | DLE (016) | Note 1 |
| 26 | [| { | ESC (027) | -1 |
| 27 |] | } | GS (029) | -1 |

Table 25. Character Codes (continued)

| KEY # | BASE CASE | UPPER CASE | CTRL | ALT |
|-------|--------------|--------------|--------------|--------------|
| 28 | CR | CR | LF (010) | -1 |
| 29 | -1 | -1 | -1 | -1 |
| 30 | a | A | SOH (001) | Note 1 |
| 31 | s | S | DC3 (019) | Note 1 |
| 32 | d | D | EOT (004) | Note 1 |
| 33 | f | F | ACK (006) | Note 1 |
| 34 | g | G | BEL (007) | Note 1 |
| 35 | h | H | BS (008) | Note 1 |
| 36 | j | J | LF (010) | Note 1 |
| 37 | k | K | VT (011) | Note 1 |
| 38 | l | L | FF (012) | Note 1 |
| 39 | : | : | -1 | -1 |
| 40 | , | , | -1 | -1 |
| 41 | ' | ' | -1 | -1 |
| 42 | -1 | -1 | -1 | -1 |
| 43 | \ | -1 | FS (028) | -1 |
| 44 | z | Z | SUB (026) | Note 1 |
| 45 | x | X | CAN (024) | Note 1 |
| 46 | c | C | ETX (003) | Note 1 |
| 47 | v | V | SYN (022) | Note 1 |
| 48 | b | B | STX (002) | Note 1 |
| 49 | n | N | SO (014) | Note 1 |
| 50 | m | M | CR (013) | Note 1 |
| 51 | , | < | -1 | -1 |
| 52 | . | > | -1 | -1 |
| 53 | / | ? | -1 | -1 |
| 54 | -1 | -1 | -1 | -1 |
| 55 | * | (Note 2) | (Note 1) | -1 |
| 56 | -1 | -1 | -1 | -1 |
| 57 | SP | SP | SP | SP |
| 58 | -1 | -1 | -1 | -1 |
| 59 | NUL (Note 1) | NUL (Note 1) | NUL (Note 1) | NUL (Note 1) |
| 60 | NUL (Note 1) | NUL (Note 1) | NUL (Note 1) | NUL (Note 1) |
| 61 | NUL (Note 1) | NUL (Note 1) | NUL (Note 1) | NUL (Note 1) |
| 62 | NUL (Note 1) | NUL (Note 1) | NUL (Note 1) | NUL (Note 1) |
| 63 | NUL (Note 1) | NUL (Note 1) | NUL (Note 1) | NUL (Note 1) |
| 64 | NUL (Note 1) | NUL (Note 1) | NUL (Note 1) | NUL (Note 1) |
| 65 | NUL (Note 1) | NUL (Note 1) | NUL (Note 1) | NUL (Note 1) |
| 66 | NUL (Note 1) | NUL (Note 1) | NUL (Note 1) | NUL (Note 1) |
| 67 | NUL (Note 1) | NUL (Note 1) | NUL (Note 1) | NUL (Note 1) |
| 68 | NUL (Note 1) | NUL (Note 1) | NUL (Note 1) | NUL (Note 1) |
| 69 | -1 | -1 | Pause | -1 |
| 70 | -1 | -1 | (Note 2) | -1 |
| 71 | -1 | -1 | Break | -1 |
| 72 | -1 | -1 | (Note 2) | -1 |

Note 1: Refer to Extended Codes Page (3-13).

Note 2: Refer to Special Handling Page (3-15).

Keys 71-83 have meaning only in base case, in NUMLOCK (or shifted) states, or in CTRL state. It should be noted that the shift key temporarily reverses the current NUMLOCK state.

| KEY # | NUM LOCK | BASE CASE | ALT | CTRL |
|-------|----------|-------------------|--------|--------------------|
| 71 | 7 | Home (Note 1) | Note 1 | Clear Screen |
| 72 | 8 | ↑ (Note 1) | Note 1 | -1 |
| 73 | 9 | PageUp (Note 1) | Note 1 | Top of Text & Home |
| 74 | - | - | -1 | -1 |
| 75 | 4 | ← (Note 1) | Note 1 | Reverse Word |
| 76 | 5 | -1 | Note 1 | (Note 1) |
| 77 | 6 | → (Note 1) | Note 1 | -1 |
| 78 | + | + | -1 | Adv Word |
| 79 | 1 | End (Note 1) | Note 1 | (Note 1) |
| 80 | 2 | ↓ (Note 1) | Note 1 | Erase to EOL |
| 81 | 3 | PageDown (Note 1) | Note 1 | (Note 1) |
| 82 | 0 | INS | Note 1 | Erase to EOS |
| 83 | | DEL (Notes 1,2) | Note 2 | (Note 1) |

Note 1: Refer to Extended Codes Page (3-13).

Note 2: Refer to Special Handling Page (3-15).

Extended Codes

A. Extended Functions

For certain functions that cannot be represented in the standard ASCII code, an extended code is used. A character code of 000 (NUL) is returned in AL. This indicates that the system or application program should examine a second code that will indicate the actual function. Usually, but not always, this second code is the scan code of the primary key that was pressed. This code is returned in AH.

Table 26. Keyboard Extended Functions

| SECOND CODE | FUNCTION |
|-------------|--|
| 3 | NUL Character |
| 15 | ← |
| 16-25 | ALT Q, W, E, R, T, Y, U, I, O, P |
| 30-38 | ALT A, S, D, F, G, H, J, K, L |
| 44-50 | ALT Z, X, C, V, B, N, M |
| 59-68 | F1-F10 Function Keys Base Case |
| 71 | Home |
| 72 | ↑ |
| 73 | Page Up & Home Cursor |
| 75 | ← → |
| 77 | End |
| 79 | ↓ |
| 80 | Page Down & Home Cursor |
| 81 | INS |
| 82 | DEL |
| 83 | |
| 84-93 | F11-F20 (Upper Case F1-F10) |
| 94-103 | F21-F30 (CTRL F1-F10) |
| 104-113 | F31-F40 (ALT F1-F10) |
| 114 | CTRL PRTSC (Start/Stop Echo to Printer) Key 55 |
| 115 | CTRL ← Reverse Word |
| 116 | CTRL → Advance Word |
| 117 | CTRL END Erase EOL |
| 118 | CTRL PG DN Erase EOS |
| 119 | CTRL HOME Clear Screen and home |
| 120-131 | ALT 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, -, = (Keys 2-13) |
| 132 | CTRL PG UP TOP 25 Lines of Text & Home Cursor |

ALT – Temporarily shifts keys 2-13, 16-25, 30-38, 44-50, and 59-68 to ALT state. Used with CTRL and DEL to cause system reset function described in Section I.3.

ALT has a special use to allow the user to enter any character code (0-255) into the system from the keyboard. The user holds down the ALT key and types the decimal value of characters using the numeric keyboard (keys 71-73, 75-77, 79-82). The ALT key is then released. If more than three digits are typed, a modulo 256 result is created. These three keys are interpreted as a character code (000-255) and are transmitted through the keyboard routine to the system or application program. ALT is handled internal to keyboard routine.

CAPS LOCK – Shifts keys 16-25, 30-38, 44-50 to upper case. A second depression of CAPS LOCK reverses the action. Handled internal to keyboard routine.

NUM LOCK – Shifts keys 71-73, 75-77, 79-83 to numeric state. A second depression of NUM LOCK reverses the action. Handled internal to keyboard routine.

SCROLL LOCK – Interpreted by appropriate application programs as indicating that the use of the cursor control keys should cause windowing over the text rather than cursor movement. A second depression of SCROLL LOCK reverses the action. The keyboard routine simply records the current shift state of SCROLL LOCK. It is up to the system or application program to perform the function.

C. Shift Key Priorities and Combinations

If combinations of ALT, CTRL and SHIFT are pressed and only one is valid, the precedence is as follows: Highest is ALT, then CTRL, then SHIFT. The only valid combination is ALT CTRL, which is used in system reset.

Special Handling

A. System Reset

The combination of ALT CTRL DEL (Key 83) will result in the keyboard routine initiating the equivalent of a system reset/reboot. Handled internal to keyboard routine.

B. Shift States

Most shift states are handled within the keyboard routine transparent to the system or application program. In any case, the current set of active shift states are available by calling an entry point in the ROM keyboard routine. The following keys result in altered shift states:

Shift – Temporarily shifts keys 2-13, 15-27, 30-41, 43-53, 55, 59-68 to upper case (lower case if in CAPSLOCK state). Temporarily reverses NUMLOCK/NUMLOCK state of keys 71-73, 75, 77, 79-83.

CTRL – Temporarily shifts keys 3, 7, 12, 14, 16-28, 30-38, 43-50, 55, 59-71, 73, 75, 77, 79, 81 to CTRL state. Used with ALT and DEL to cause "system reset" function described in Section I.3. Used with SCROLL LOCK to cause "break" function described in Section I.3. Used with NUMLOCK to cause "pause" function described in Section I.3.

B. Break

The combination CTRL BREAK will result in the keyboard routine signaling interrupt -1A. Also, the extended characters (AL = 00H, AH = 00H) will be returned.

Power up initialization, this interrupt is set up to cause the break sequence to be ignored. It is up to the system or application initialization code to change the interrupt vector in order to support an actual "break" function.

C. Pause

The combination CTRL NUM-LOCK will cause the keyboard interrupt routine to loop, waiting for any key except NUM-LOCK to be pressed. This provides a system/application transparent method of suspending list/print/etc. temporarily, and then resuming. The "Unpause" key is thrown away. Handled internal to keyboard routine.

The following keys will have their typematic action suppressed by the keyboard routine: CTRL, SHIFT, ALT, NUM-LOCK, SCROLL-LOCK, CAPS LOCK, INS.

E. Print Screen

The combination SHIFT-PRINT SCREEN (Key 55) will result in an interrupt invoking the print screen routine. This routine works in alpha/graphics mode, with unrecognizable characters printing as blanks.

The keyboard routine does its own buffering. The buffer is big enough to support a fast typist. If a key is entered when the buffer is full, the key will be ignored and the "bell" will be sounded.

Keyboard Usage

This section is intended to outline a set of guidelines for key usage when performing commonly used functions.

Table 27. Keyboard - Commonly Used Functions

| FUNCTION | KEY(S) | COMMENT |
|--|--------------------|------------------------------------|
| Home Cursor | HOME | Editors; word processors |
| Return to outermost menu | HOME | Menu driven applications |
| Move cursor up | ↑ | Full screen editor, word processor |
| Page up, scroll backwards 25 lines & home | PG UP | Editors; word processors |
| Move cursor left | ← Key 75 | Text, command entry |
| Move cursor right | → | Text, command entry |
| Scroll to end of text Place cursor at end of line | END | Editors; word processors |
| Move cursor down | ↓ | Full screen editor, word processor |
| Page down, scroll forwards 25 lines & home | PG DN | Editors; word processors |
| Start/Stop insert text at cursor, shift text right in buffer | INS | Text, command entry |
| Delete character at cursor | DEL | Text, command entry |
| Destructive backspace | ← Key 14 | Text, command entry |
| Tab forward | → | Text entry |
| Tab reverse | ← | Text entry |
| Clear screen and home | CTRL HOME | Command entry |
| Scroll up | ↑ | In scroll lock mode |
| Scroll down | ↓ | In scroll lock mode |
| Scroll left | ← | In scroll lock mode |
| Scroll right | → | In scroll lock mode |
| Delete from cursor to EOL | CTRL END | Text, command entry |
| Exit/Escape | ESC | Editor, 1 level of menu, etc |
| Start/Stop Echo screen to printer | PRATSC CTRL K55 | Any time |
| Delete from cursor to EOS | CTRL PG DN | Text, command entry |

Table 27. Keyboard – Commonly Used Functions (continued)

| FUNCTION | KEY(S) | COMMENT |
|---------------------------------|---|---|
| Advance word | CTRL → | Text entry |
| Reverse word | CTRL ← | Text entry |
| Window Right | CTRL → | When text is too wide to fit screen |
| Window Left | CTRL ← | When text is too wide to fit screen |
| Enter insert mode | INS | Line editor |
| Exit insert mode | INS | Line editor |
| Cancel current line | ESC | Command entry, text entry |
| Suspend system (pause) | CTRL NUMLOCK | Stop list, stop program, etc. Resumes on any key |
| Break interrupt | CTRL BREAK | Interrupt current process |
| System reset | ALT CTRL DEL | Reboot |
| Top of document and home cursor | CTRL PG UP | Editors, word processors |
| Standard Function Keys | F1–F10 | Primary function keys |
| Secondary function keys | SHIFT F1–F10 CTRL F1–F10 ALT F1–F10 | Extra function keys if 10 are not sufficient |
| Extra function keys | ALT Keys 2–13 (1–9, 0, –, =) | Used when stickers are put along top of keyboard |
| Extra function keys | ALT A–Z | Used when function starts with same letter as one of the alpha keys |

Table 28. BASIC Screen Editor Special Functions

| FUNCTION | KEY |
|--------------------------------|--------------|
| Carriage return | ↵ |
| Line feed | CTRL ↵ |
| Bell | CTRL G |
| Home | HOME |
| Cursor up | ↑ |
| Cursor down | ↓ |
| Cursor left | ← |
| Cursor right | → |
| Advance one word | CTRL → |
| Reverse one word | CTRL ← |
| Insert | INS |
| Delete | DEL |
| Clear screen | CTRL HOME |
| Freeze output | CTRL NUMLOCK |
| Tab advance | → |
| Stop execution (break) | CTRL BREAK |
| Delete current line | ESC |
| Delete to end of line | CTRL END |
| Position cursor to end of line | END |

ROM

Table 29. DOS Special Functions

| FUNCTION | KEY |
|-------------------------------|------------------------------------|
| Suspend | CTRL NUMLOCK |
| Echo to printer | CTRL-PRN |
| Stop echo to printer | (Key 55 any case) CTRL-PRN |
| Exit current function (break) | (Key 55 any case) CTRL BREAK |
| Backspace | ← Key 14 |
| Line feed | CTRL ↵ |
| Cancel line | ESC |
| Copy character | F1 or → |
| Copy till match | F2 |
| Copy remaining | F3 |
| Skip character | DEL |
| Skip until match | F4 |
| Enter insert mode | INS |
| Exit insert mode | INS |
| Make new line the template | F5 |
| String separator in REPLACE | F6 |
| End of file in keyboard input | F6 |

Low Memory Maps (0-0600'x)

ROM

Table 30. Interrupt Vectors (0-7F)

| ADDRESS HEX | INTERRUPT HEX | FUNCTION |
|----------------|------------------|--|
| 0-3 | 0 | Divide by Zero |
| 4-7 | 1 | Single step |
| 8-B | 2 | Non-Maskable Interrupt (NMI) |
| C-F | 3 | Break Point Instruction ('CC'x) |
| 10-13 | 4 | Overflow |
| 14-17 | 5 | Print Screen |
| 18-1F | 6,7 | Reserved |
| 20-23 | 8 | Timer (18.2 per second) |
| 24-27 | 9 | Keyboard Interrupt |
| 28-37 | A,B,C,D | Reserved |
| 38-3B | E | Diskette Interrupt |
| 3C-3F | F | Reserved |
| 40-43 | 10 | Video I/O Call |
| 44-47 | 11 | Equipment Check Call |
| 48-4B | 12 | Memory Check Call |
| 4C-4F | 13 | Diskette I/O Call |
| 50-53 | 14 | RS232 I/O Call |
| 54-57 | 15 | Cassette I/O Call |
| 58-5B | 16 | Keyboard I/O Call |
| 5C-5F | 17 | Printer I/O Call |
| 60-63 | 18 | ROM Basic Entry Code |
| 64-67 | 19 | Boot Strap Loader |
| 68-6B | 1A | Time of Day Call |
| 6C-6F | 1B | Get Control on Keyboard Break: Note 1 |
| 70-73 | 1C | Get Control on timer interrupt: Note 1 |
| 74-77 | 1D | Pointer to video initialization table: Note 2 |
| 78-7B | 1E | Pointer to diskette parameter table: Note 2 |
| 7C-7F | 1F | Pointer to table (1KB) for graphics character Generator for ASCII 128-255. Defaults to 0:0 |
| Notes: | | (1) Initialized at power up to point to an IRET instruction. (2) Initialized at power up to point to tables in ROM. |

Table 31. BASIC & DOS Reserved Interrupts (80-3FF)

| ADDRESS HEX | INTERERRUPT HEX | FUNCTION |
|----------------|--------------------|---|
| 80-83 | 20 | DOS Program Terminate |
| 84-87 | 21 | DOS Function Call |
| 88-8B | 22 | DOS Terminate Address |
| 8C-8F | 23 | DOS CTRL-BRK Exit Address |
| 90-93 | 24 | DOS Fatal Error Vector |
| 94-97 | 25 | DOS Absolute Disk read |
| 98-9B | 26 | DOS Absolute Disk write |
| 9C-9F | 27 | DOS Terminate, Fix in Storage |
| A0-FF | 28-3F | Reserved for DOS |
| 100-1FF | 40-7F | Not Used |
| 200-217 | 80-85 | Reserved By BASIC |
| 218-3C3 | 86-F0 | Used by BASIC Interpreter while BASIC is Running. |
| 3C4-3FF | F1-FF | Not Used |

Table 32. Reserved Memory Locations (400-5FF)

| ADDRESS HEX | MDDE | FUNCTION |
|----------------|----------|--|
| 400-48F | ROM BIOS | See BIOS Listing |
| 490-4CF | DOS | Used by DOS Mode Command |
| 4D0-4EF | | Reserved |
| 4F0-4FF | | Reserved as Intra-Application Communication area for any application. |
| | | Reserved for DOS and BASIC |
| 500-5FF | | Print Screen status flag store. |
| 500 | DOS | 0-Print screen not active or successful print screen operation. 1-Print screen in progress. |
| | | 255-Error encountered during print screen operation. |
| 504 | DOS | Single drive mode status byte. |
| 510-511 | BASIC | BASIC's segment address store. |
| 512-515 | BASIC | Clock interrupt vector segment: offset store. |
| 516-519 | BASIC | Break key interrupt vector segment: offset store. |
| 51A-51D | BASIC | Disk error interrupt vector segment: offset store. |

BASIC Workspace Variables

If you do DEF SEG (Default workspace segment)

| | OFFSET | LENGTH |
|---|---------|--------|
| Line number of current line being executed | X '2E' | 2 |
| Line number of last error | X '347' | 2 |
| Offset into segment of start of program text | X '30' | 2 |
| Offset into of start of variables (end of program text 1-1) | X '358' | 2 |
| Keyboard buffer contents if 0-no rharacters in buffer if 1-characters in buffer if you POKE & H6A, 0 you flush any characters in buffer | X '6A' | 1 |

Example:

100 Print PEEK (&H2E) + 256*PEEK (&H2F)

100

L X '64'

H X '00'

Contents

Appendix A: ROM BIOS Listing

Appendix B: Assembly Instruction Set Reference

Appendix C: Of Characters, Keystrokes and Color.

Appendix D: Logic Diagrams

Appendix E: Unit Specifications

Contents

| ITEM | LINE NUMBER | ADDRESS | PAGE |
|-------------------------------|----------------|---------|------|
| Equates and Data Areas | 1 | --- | A-2 |
| Power-on Self-test | 198 | E016 | A-4 |
| Boot Strap Loader | 1355 | E6F2 | A-20 |
| I/O Support | | | |
| Asynchronous | | | |
| Communications | | | |
| (RS 232) I/O | 1410 | E729 | A-20 |
| Keyboard I/O | 1640 | E82E | A-23 |
| Diskette I/O | 2255 | EC59 | A-32 |
| Printer I/O | 3007 | EFD2 | A-42 |
| Display (VIDEO) I/O | 3119 | F045 | A-43 |
| Cassette I/O | 4977 | F859 | A-68 |
| System Configuration Analysis | | | |
| Memory-Size- | | | |
| Determination | 4903 | F841 | A-67 |
| Equipment- | | | |
| Determination | | | |
| (Options) | 4933 | F84D | A-67 |
| Graphics-Character Generator | 5503 | FA6E | A-75 |
| Time-of-day | 5642 | FE6E | A-77 |
| Print Screen | 5817 | FF53 | A-79 |
| Notes for the BIOS Listing | | | A-81 |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|---------|------|--|---------|------|---|
| | 1 | ***** BIOS FOR IBM PERSONAL COMPUTER ***** | | 78 | CTL_SHIFT EQU 04H ; CONTROL SHIFT KEY DEPRESSED |
| | 2 | 1 EQUATES | | 79 | LEFT_SHIFT EQU 02H ; LEFT SHIFT KEY DEPRESSED |
| | 3 | 1 | | 80 | RIGHT_SHIFT EQU 01H ; RIGHT SHIFT KEY DEPRESSED |
| 0000 | 4 | PORT_A EQU 60H ; 16555 PORT A ADDR | | 81 | 1 |
| 0001 | 5 | PORT_B EQU 61H ; 16555 PORT B ADDR | | 82 | KB_FLAG_1 DB ? ; SECOND BYTE OF KEYBOARD STATUS |
| 0002 | 6 | PORT_C EQU 62H ; 16555 PORT C ADDR | | 83 | 1 |
| 0003 | 7 | PORT_D EQU 63H ; 16555 PORT D ADDR | | 84 | INS_SHIFT EQU 00H ; INSERT KEY IS DEPRESSED |
| 0004 | 8 | PORT_E EQU 64H ; 16555 PORT E ADDR | | 85 | CAPS_LOCK EQU 00H ; CAPS LOCK KEY IS DEPRESSED |
| 0005 | 9 | PORT_F EQU 65H ; 16555 PORT F ADDR | | 86 | NUM_LOCK EQU 00H ; NUM LOCK KEY IS DEPRESSED |
| 0006 | 10 | PORT_G EQU 66H ; 16555 PORT G ADDR | | 87 | SCROLL_LOCK EQU 00H ; SCROLL LOCK KEY IS DEPRESSED |
| 0007 | 11 | PORT_H EQU 67H ; 16555 PORT H ADDR | | 88 | HOLD_STATE EQU 00H ; SUSPEND KEY HAS BEEN TOGGLED |
| 0008 | 12 | PORT_I EQU 68H ; 16555 PORT I ADDR | | 89 | 1 |
| 0009 | 13 | PORT_J EQU 69H ; 16555 PORT J ADDR | | 90 | ALT_INPUT DB ? ; STORAGE FOR ALTERNATE KEYPAD ENTRY |
| 0010 | 14 | PORT_K EQU 70H ; 16555 PORT K ADDR | | 91 | BUFFER_HEAD DB ? ; POINTER TO HEAD OF KEYBOARD BUFFER |
| 0011 | 15 | PORT_L EQU 71H ; 16555 PORT L ADDR | | 92 | BUFFER_TAIL DB ? ; POINTER TO TAIL OF KEYBOARD BUFFER |
| 0012 | 16 | PORT_M EQU 72H ; 16555 PORT M ADDR | | 93 | KB_BUFFER DB 16 DUP(1) ; ROOM FOR 15 ENTRIES |
| 0013 | 17 | PORT_N EQU 73H ; 16555 PORT N ADDR | | 94 | KB_BUFFER_END LABEL WORD |
| 0014 | 18 | PORT_O EQU 74H ; 16555 PORT O ADDR | | 95 | 1 |
| 0015 | 19 | PORT_P EQU 75H ; 16555 PORT P ADDR | | 96 | 1 |
| 0016 | 20 | PORT_Q EQU 76H ; 16555 PORT Q ADDR | | 97 | 1 |
| 0017 | 21 | PORT_R EQU 77H ; 16555 PORT R ADDR | | 98 | NUM_LOCK EQU 69 ; SCAN CODE FOR NUMBER LOCK |
| 0018 | 22 | PORT_S EQU 78H ; 16555 PORT S ADDR | | 99 | SCROLL_LOCK EQU 70 ; SCROLL LOCK KEY |
| 0019 | 23 | PORT_T EQU 79H ; 16555 PORT T ADDR | | 100 | ALT_KEY EQU 56 ; ALTERNATE SHIFT KEY SCAN CODE |
| 0020 | 24 | PORT_U EQU 80H ; 16555 PORT U ADDR | | 101 | CTL_KEY EQU 29 ; SCAN CODE FOR CONTROL KEY |
| 0021 | 25 | PORT_V EQU 81H ; 16555 PORT V ADDR | | 102 | CAPS_KEY EQU 58 ; SCAN CODE FOR SHIFT LOCK |
| 0022 | 26 | PORT_W EQU 82H ; 16555 PORT W ADDR | | 103 | LEFT_KEY EQU 42 ; SCAN CODE FOR LEFT SHIFT |
| 0023 | 27 | PORT_X EQU 83H ; 16555 PORT X ADDR | | 104 | RIGHT_KEY EQU 54 ; SCAN CODE FOR RIGHT SHIFT |
| 0024 | 28 | PORT_Y EQU 84H ; 16555 PORT Y ADDR | | 105 | INS_KEY EQU 62 ; SCAN CODE FOR INSERT KEY |
| 0025 | 29 | PORT_Z EQU 85H ; 16555 PORT Z ADDR | | 106 | DEL_KEY EQU 63 ; SCAN CODE FOR DELETE KEY |
| 0026 | 30 | PORT_0 EQU 86H ; 16555 PORT 0 ADDR | | 107 | 1 |
| 0027 | 31 | PORT_1 EQU 87H ; 16555 PORT 1 ADDR | | 108 | 1 |
| 0028 | 32 | PORT_2 EQU 88H ; 16555 PORT 2 ADDR | | 109 | 1 |
| 0029 | 33 | PORT_3 EQU 89H ; 16555 PORT 3 ADDR | | 110 | 1 |
| 0030 | 34 | PORT_4 EQU 90H ; 16555 PORT 4 ADDR | | 111 | 1 |
| 0031 | 35 | PORT_5 EQU 91H ; 16555 PORT 5 ADDR | | 112 | 1 |
| 0032 | 36 | PORT_6 EQU 92H ; 16555 PORT 6 ADDR | | 113 | 1 |
| 0033 | 37 | PORT_7 EQU 93H ; 16555 PORT 7 ADDR | | 114 | INT_FLAG EQU 00H ; INTERRUPT OCCURRENCE FLAG |
| 0034 | 38 | PORT_8 EQU 94H ; 16555 PORT 8 ADDR | | 115 | 1 |
| 0035 | 39 | PORT_9 EQU 95H ; 16555 PORT 9 ADDR | | 116 | 1 |
| 0036 | 40 | PORT_A EQU 96H ; 16555 PORT A ADDR | | 117 | 1 |
| 0037 | 41 | PORT_B EQU 97H ; 16555 PORT B ADDR | | 118 | 1 |
| 0038 | 42 | PORT_C EQU 98H ; 16555 PORT C ADDR | | 119 | 1 |
| 0039 | 43 | PORT_D EQU 99H ; 16555 PORT D ADDR | | 120 | 1 |
| 0040 | 44 | PORT_E EQU 00H ; 16555 PORT E ADDR | | 121 | 1 |
| 0041 | 45 | PORT_F EQU 01H ; 16555 PORT F ADDR | | 122 | 1 |
| 0042 | 46 | PORT_G EQU 02H ; 16555 PORT G ADDR | | 123 | 1 |
| 0043 | 47 | PORT_H EQU 03H ; 16555 PORT H ADDR | | 124 | 1 |
| 0044 | 48 | PORT_I EQU 04H ; 16555 PORT I ADDR | | 125 | 1 |
| 0045 | 49 | PORT_J EQU 05H ; 16555 PORT J ADDR | | 126 | 1 |
| 0046 | 50 | PORT_K EQU 06H ; 16555 PORT K ADDR | | 127 | 1 |
| 0047 | 51 | PORT_L EQU 07H ; 16555 PORT L ADDR | | 128 | 1 |
| 0048 | 52 | PORT_M EQU 08H ; 16555 PORT M ADDR | | 129 | 1 |
| 0049 | 53 | PORT_N EQU 09H ; 16555 PORT N ADDR | | 130 | 1 |
| 0050 | 54 | PORT_O EQU 0AH ; 16555 PORT O ADDR | | 131 | 1 |
| 0051 | 55 | PORT_P EQU 0BH ; 16555 PORT P ADDR | | 132 | 1 |
| 0052 | 56 | PORT_Q EQU 0CH ; 16555 PORT Q ADDR | | 133 | 1 |
| 0053 | 57 | PORT_R EQU 0DH ; 16555 PORT R ADDR | | 134 | 1 |
| 0054 | 58 | PORT_S EQU 0EH ; 16555 PORT S ADDR | | 135 | 1 |
| 0055 | 59 | PORT_T EQU 0FH ; 16555 PORT T ADDR | | 136 | 1 |
| 0056 | 60 | PORT_U EQU 10H ; 16555 PORT U ADDR | | 137 | 1 |
| 0057 | 61 | PORT_V EQU 11H ; 16555 PORT V ADDR | | 138 | 1 |
| 0058 | 62 | PORT_W EQU 12H ; 16555 PORT W ADDR | | 139 | 1 |
| 0059 | 63 | PORT_X EQU 13H ; 16555 PORT X ADDR | | 140 | 1 |
| 0060 | 64 | PORT_Y EQU 14H ; 16555 PORT Y ADDR | | 141 | 1 |
| 0061 | 65 | PORT_Z EQU 15H ; 16555 PORT Z ADDR | | 142 | 1 |
| 0062 | 66 | PORT_0 EQU 16H ; 16555 PORT 0 ADDR | | 143 | 1 |
| 0063 | 67 | PORT_1 EQU 17H ; 16555 PORT 1 ADDR | | 144 | 1 |
| 0064 | 68 | PORT_2 EQU 18H ; 16555 PORT 2 ADDR | | 145 | 1 |
| 0065 | 69 | PORT_3 EQU 19H ; 16555 PORT 3 ADDR | | 146 | 1 |
| 0066 | 70 | PORT_4 EQU 1AH ; 16555 PORT 4 ADDR | | 147 | 1 |
| 0067 | 71 | PORT_5 EQU 1BH ; 16555 PORT 5 ADDR | | 148 | 1 |
| 0068 | 72 | PORT_6 EQU 1CH ; 16555 PORT 6 ADDR | | 149 | 1 |
| 0069 | 73 | PORT_7 EQU 1DH ; 16555 PORT 7 ADDR | | 150 | 1 |
| 0070 | 74 | PORT_8 EQU 1EH ; 16555 PORT 8 ADDR | | 151 | 1 |
| 0071 | 75 | PORT_9 EQU 1FH ; 16555 PORT 9 ADDR | | 152 | 1 |
| 0072 | 76 | PORT_A EQU 20H ; 16555 PORT A ADDR | | 153 | 1 |
| 0073 | 77 | PORT_B EQU 21H ; 16555 PORT B ADDR | | 154 | 1 |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|-------------------|------|---|-------------|------|--------------|
| 0068 ?? | 155 | LAST_VAL DB ? ; LAST INPUT VALUE | E02A | 229 | DEC C3; |
| | 156 | ; | E024 4F | 230 | DEC D1 |
| | 157 | ; | E020 7F | 231 | MOV SI,D1 |
| | 158 | ; | E02C 8877 | 232 | MOV CX,BX |
| 006C 7777 | 159 | ; | E02E 88C8 | 233 | MOV LOD58 |
| 006E 7777 | 160 | TIMER_LOW DB ? ; LOW WORD OF TIMER COUNT | E030 AC | 234 | XOR AL,AH |
| 0070 77 | 161 | TIMER_HIGH DB ? ; HIGH WORD OF TIMER COUNT | E031 32C4 | 235 | XOR C7 |
| | 162 | TIMER_OFL DB ? ; TIMER HAS ROLLED OVER SINCE LAST READ | E033 7525 | 236 | JNE C7 |
| | 163 | COUNTS_SEC DB ? ; | E035 E442 | 237 | IN AL,PORT_C |
| | 164 | COUNTS_MIN DB ? ; | E037 24C0 | 238 | AND AL,0C0H |
| | 165 | COUNTS_HOUR DB ? ; | E039 8000 | 239 | MOV AL,0 |
| | 166 | COUNTS_DAY DB ? ; | E03B 7510 | 240 | JNZ C7 |
| | 167 | ; | E03D 80FC00 | 241 | CMP AH,0 |
| | 168 | ; | E040 7463 | 242 | JE C6 |
| | 169 | ; | E042 84C2 | 243 | MOV AL,DL |
| | 170 | ; | E044 AA | 244 | STOSB |
| 0071 77 | 171 | BIOS_BREAK DB ? ; BIT 7 = 1 IF BREAK KEY HAS BEEN DEPRESSED | E046 E2E9 | 245 | LOOP C5 |
| 0072 7777 | 172 | RESET_FLAG DB ? ; WORD = 1234H IF KEYBOARD RESET INTERNAL | E047 80FC00 | 246 | CHP |
| | 173 | DATA ENDS | E04A 740E | 247 | CHP AH,0 |
| | 174 | ; | E04C 84E0 | 248 | JE C7 |
| | 175 | ; | E04E 88F2 | 249 | MOV AH,AL |
| 0050 | 176 | ; | E050 FC | 250 | XCHG DH,DL |
| 0000 77 | 177 | ; | E051 47 | 251 | CLO |
| | 178 | ; | E052 740B | 252 | JNC D1 |
| | 179 | STATUS_BYTE DB ? | E053 47 | 253 | JZ C4 |
| | 180 | ; | E054 4F | 254 | DEC D1 |
| | 181 | ; | E055 8A0100 | 255 | MOV DX,1 |
| | 182 | ; | E056 E800 | 256 | MOV DX,1 |
| | 183 | ; | E05A | 257 | JMP SHORT C3 |
| 0000 | 184 | ; | E05A C3 | 258 | RET |
| 0000 | 185 | VIDEO_RAM SEGMENT AT 0B00H | E05B FA | 259 | STOSB |
| 0000 | 186 | REGEND LABEL BYTE | E05C 84D5 | 260 | ENDP |
| 0000 | 187 | REGEND LABEL WORD | E05E 9E | 261 | TEST D1 |
| 0000 (1630A 77) | 188 | REGEND LABEL WORD | E05F 754E | 262 | ; |
| | 189 | VIDEO_RAM ENDS | E061 754C | 263 | ; |
| | 190 | ; | E063 7B6A | 264 | ; |
| | 191 | ; | E065 7F4B | 265 | ; |
| | 192 | ; | E067 9F | 266 | ; |
| | 193 | ; | E068 B105 | 267 | ; |
| 0000 (5730A 77) | 194 | CODE SEGMENT AT 0F00H | E06A D2EC | 268 | ; |
| | 195 | DB 5730A DUP(?) ; FILL LOWEST 50K | E06C 7341 | 269 | ; |
| 0000 (5730A 77) | 196 | DB '5700551 COMP. IBM 1981' ; COPYRIGHT NOTICE | E06E 8B40 | 270 | ; |
| 0000 357730303531 | 197 | ; | E070 0B00 | 271 | ; |
| 20034F5052E20 | 198 | ; | E072 713B | 272 | ; |
| 49424020313926 | 199 | ; | E074 32E4 | 273 | ; |
| 31 | 200 | ; | E076 9E | 274 | ; |
| | 201 | ; | E077 7136 | 275 | ; |
| | 202 | ; | E079 7434 | 276 | ; |
| | 203 | ; | E07B 7832 | 277 | ; |
| | 204 | ; | E07D 7430 | 278 | ; |
| 0016 0000 | 205 | ; | E07F 9F | 279 | ; |
| 0018 0001 | 206 | ; | E080 B105 | 280 | ; |
| | 207 | ; | E082 D2EC | 281 | ; |
| | 208 | ; | E084 7329 | 282 | ; |
| | 209 | ; | E086 00E4 | 283 | ; |
| | 210 | ; | E088 7025 | 284 | ; |
| | 211 | ; | E08A 8877FF | 285 | ; |
| | 212 | ; | E08C 80D8 | 286 | ; |
| | 213 | ; | E08E 8C08 | 287 | ; |
| | 214 | ; | E090 8C08 | 288 | ; |
| | 215 | ; | E092 8C08 | 289 | ; |
| | 216 | ; | E094 8C08 | 290 | ; |
| | 217 | ; | E096 8C08 | 291 | ; |
| | 218 | ; | E098 8C08 | 292 | ; |
| | 219 | ; | E09A 8C08 | 293 | ; |
| 001A 0000 | 220 | STOSB | E09C 8C08 | 294 | ; |
| 001B 0000 | 221 | MOV CX,4000H | E09E 8C08 | 295 | ; |
| 001C 0000 | 222 | ; | E0A0 8C08 | 296 | ; |
| 001E 0000 | 223 | ; | E0A2 8C08 | 297 | ; |
| 0020 0000FF | 224 | ; | E0A4 8C08 | 298 | ; |
| 0023 8455AA | 225 | ; | E0A6 8C08 | 299 | ; |
| 0026 0000 | 226 | ; | E0A8 8C08 | 300 | ; |
| 0028 F3 | 227 | ; | E0AA 8C08 | 301 | ; |
| 0029 AA | 228 | ; | E0AC 8C08 | 302 | ; |
| | 229 | ; | E0AE 8C08 | 303 | ; |
| | 230 | ; | E0B0 8C08 | 304 | ; |
| | 231 | ; | E0B2 8C08 | 305 | ; |
| | 232 | ; | E0B4 8C08 | 306 | ; |

| LOC OBJ | LINE | SOURCE |
|-----------|------|--|
| E000 88F5 | 307 | MOV SI,BP |
| E000 88FE | 308 | JNC C9 |
| E000 8907 | 309 | JNC C9 |
| E000 8910 | 310 | XOP AX,DI |
| E000 8919 | 311 | JNZ E8B0 |
| E000 8922 | 312 | CLC |
| E000 892B | 313 | JNC C8 |
| E000 8934 | 314 | C9: |
| E000 8935 | 315 | OP AX,DI |
| E000 8936 | 316 | JZ C10 |
| E000 8937 | 317 | ERR01: JLT |
| E000 8938 | 318 | TEST 02 |
| E000 8939 | 319 | ROD CHECKSUM TEST 1 |
| E000 893A | 320 | DESCRIPTION |
| E000 893B | 321 | A CHECKSUM IS DONE FOR THE OK ROD MODULE CONTAINING P00 AND BIOS. |
| E000 893C | 322 | DESCRIPTION |
| E000 893D | 323 | C10: |
| E000 893E | 324 | MOV AL,0 |
| E000 893F | 325 | OUT 0A0H,AL |
| E000 8940 | 326 | OUT 03H,AL |
| E000 8941 | 327 | MOV AL,7FH |
| E000 8942 | 328 | OUT 03H,AL |
| E000 8943 | 329 | OUT 03H,AL |
| E000 8944 | 330 | MOV AL,0FH |
| E000 8945 | 331 | OUT 03H,AL |
| E000 8946 | 332 | SUB AL,AL |
| E000 8947 | 333 | MOV DX,30BH |
| E000 8948 | 334 | OUT DX,AL |
| E000 8949 | 335 | INC AL |
| E000 894A | 336 | MOV DX,30BH |
| E000 894B | 337 | CUT DX,AL |
| E000 894C | 338 | MOV AX,CODE |
| E000 894D | 339 | MOV SS,AX |
| E000 894E | 340 | MOV DX,0E00H |
| E000 894F | 341 | MOV SP,OFFSET C1 |
| E000 8950 | 342 | JIP POS CHECKSUM |
| E000 8951 | 343 | C11: JUE ERR01 |
| E000 8952 | 344 | TEST 03 |
| E000 8953 | 345 | 8337 DMA INITIALIZATION CHANNEL REGISTER TEST |
| E000 8954 | 346 | DESCRIPTION |
| E000 8955 | 347 | DISABLE THE 8337 DMA CONTROLLER. VERIFY THAT TIMER 1 FUNCTIONS OK. |
| E000 8956 | 348 | WRITE/READ THE CURRENT ADDRESS AND WORD COUNT REGISTERS FOR ALL CHANNELS. INITIALIZE AND START DMA FOR MEMORY REFRESH. |
| E000 8957 | 349 | DISABLE DMA CONTROLLER |
| E000 8958 | 350 | MOV AL,04 |
| E000 8959 | 351 | OUT 0A0H,AL |
| E000 895A | 352 | VERIFY THAT TIMER 1 FUNCTIONS OK |
| E000 895B | 353 | MOV AL,54H |
| E000 895C | 354 | OUT 0A0H,AL |
| E000 895D | 355 | MOV CX,CX |
| E000 895E | 356 | SUB BL,CL |
| E000 895F | 357 | MOV AL,CL |
| E000 8960 | 358 | OUT 0A0H,AL |
| E000 8961 | 359 | MOV AL,40H |
| E000 8962 | 360 | OUT 0A0H,AL |
| E000 8963 | 361 | IN AL,TIMER+1 |
| E000 8964 | 362 | OR BL,AL |
| E000 8965 | 363 | CMP BL,0FH |
| E000 8966 | 364 | JC C12 |
| E000 8967 | 365 | LOOP |
| E000 8968 | 366 | JMP SHORT ERR01 |
| E000 8969 | 367 | MOV AL,04 |
| E000 896A | 368 | OUT 0A0H,AL |
| E000 896B | 369 | MOV CX,CX |
| E000 896C | 370 | OUT 0A0H,AL |
| E000 896D | 371 | MOV AL,40H |
| E000 896E | 372 | OUT 0A0H,AL |
| E000 896F | 373 | MOV AL,40H |
| E000 8970 | 374 | OUT 0A0H,AL |
| E000 8971 | 375 | MOV AL,40H |
| E000 8972 | 376 | OUT 0A0H,AL |
| E000 8973 | 377 | MOV AL,40H |
| E000 8974 | 378 | OUT 0A0H,AL |
| E000 8975 | 379 | MOV AL,40H |
| E000 8976 | 380 | OUT 0A0H,AL |
| E000 8977 | 381 | IN AL,TIMER+1 |
| E000 8978 | 382 | OR BL,AL |
| E000 8979 | 383 | JC C13 |
| E000 897A | 384 | LOOP |
| E000 897B | 385 | MOV SI,BP |
| E000 897C | 386 | INITIALIZE TIMER 1 TO REFRESH MEMORY |
| E000 897D | 387 | MOV AL,54H |
| E000 897E | 388 | OUT 0A0H,AL |
| E000 897F | 389 | MOV AL,16 |
| E000 8980 | 390 | OUT 0A0H,AL |
| E000 8981 | 391 | MOV AL,16 |
| E000 8982 | 392 | OUT 0A0H,AL |
| E000 8983 | 393 | MOV AL,16 |
| E000 8984 | 394 | OUT 0A0H,AL |
| E000 8985 | 395 | MOV AL,16 |
| E000 8986 | 396 | OUT 0A0H,AL |
| E000 8987 | 397 | MOV AL,16 |
| E000 8988 | 398 | OUT 0A0H,AL |
| E000 8989 | 399 | MOV AL,16 |
| E000 898A | 400 | OUT 0A0H,AL |
| E000 898B | 401 | MOV AL,16 |
| E000 898C | 402 | OUT 0A0H,AL |
| E000 898D | 403 | MOV AL,16 |
| E000 898E | 404 | OUT 0A0H,AL |
| E000 898F | 405 | MOV AL,16 |
| E000 8990 | 406 | OUT 0A0H,AL |
| E000 8991 | 407 | MOV AL,16 |
| E000 8992 | 408 | OUT 0A0H,AL |
| E000 8993 | 409 | MOV AL,16 |
| E000 8994 | 410 | OUT 0A0H,AL |
| E000 8995 | 411 | MOV AL,16 |
| E000 8996 | 412 | OUT 0A0H,AL |
| E000 8997 | 413 | MOV AL,16 |
| E000 8998 | 414 | OUT 0A0H,AL |
| E000 8999 | 415 | MOV AL,16 |
| E000 899A | 416 | OUT 0A0H,AL |
| E000 899B | 417 | MOV AL,16 |
| E000 899C | 418 | OUT 0A0H,AL |
| E000 899D | 419 | MOV AL,16 |
| E000 899E | 420 | OUT 0A0H,AL |
| E000 899F | 421 | MOV AL,16 |
| E000 89A0 | 422 | OUT 0A0H,AL |
| E000 89A1 | 423 | MOV AL,16 |
| E000 89A2 | 424 | OUT 0A0H,AL |
| E000 89A3 | 425 | MOV AL,16 |
| E000 89A4 | 426 | OUT 0A0H,AL |
| E000 89A5 | 427 | MOV AL,16 |
| E000 89A6 | 428 | OUT 0A0H,AL |
| E000 89A7 | 429 | MOV AL,16 |
| E000 89A8 | 430 | OUT 0A0H,AL |
| E000 89A9 | 431 | MOV AL,16 |
| E000 89AA | 432 | OUT 0A0H,AL |
| E000 89AB | 433 | MOV AL,16 |
| E000 89AC | 434 | OUT 0A0H,AL |
| E000 89AD | 435 | MOV AL,16 |
| E000 89AE | 436 | OUT 0A0H,AL |
| E000 89AF | 437 | MOV AL,16 |
| E000 89B0 | 438 | OUT 0A0H,AL |
| E000 89B1 | 439 | MOV AL,16 |
| E000 89B2 | 440 | OUT 0A0H,AL |
| E000 89B3 | 441 | MOV AL,16 |
| E000 89B4 | 442 | OUT 0A0H,AL |
| E000 89B5 | 443 | MOV AL,16 |
| E000 89B6 | 444 | OUT 0A0H,AL |
| E000 89B7 | 445 | MOV AL,16 |
| E000 89B8 | 446 | OUT 0A0H,AL |
| E000 89B9 | 447 | MOV AL,16 |
| E000 89BA | 448 | OUT 0A0H,AL |
| E000 89BB | 449 | MOV AL,16 |
| E000 89BC | 450 | OUT 0A0H,AL |
| E000 89BD | 451 | MOV AL,16 |
| E000 89BE | 452 | OUT 0A0H,AL |
| E000 89BF | 453 | MOV AL,16 |
| E000 89C0 | 454 | OUT 0A0H,AL |
| E000 89C1 | 455 | MOV AL,16 |
| E000 89C2 | 456 | OUT 0A0H,AL |
| E000 89C3 | 457 | MOV AL,16 |
| E000 89C4 | 458 | OUT 0A0H,AL |
| E000 89C5 | 459 | MOV AL,16 |
| E000 89C6 | 460 | OUT 0A0H,AL |
| E000 89C7 | 461 | MOV AL,16 |
| E000 89C8 | 462 | OUT 0A0H,AL |
| E000 89C9 | 463 | MOV AL,16 |
| E000 89CA | 464 | OUT 0A0H,AL |
| E000 89CB | 465 | MOV AL,16 |
| E000 89CC | 466 | OUT 0A0H,AL |
| E000 89CD | 467 | MOV AL,16 |
| E000 89CE | 468 | OUT 0A0H,AL |
| E000 89CF | 469 | MOV AL,16 |
| E000 89D0 | 470 | OUT 0A0H,AL |
| E000 89D1 | 471 | MOV AL,16 |
| E000 89D2 | 472 | OUT 0A0H,AL |
| E000 89D3 | 473 | MOV AL,16 |
| E000 89D4 | 474 | OUT 0A0H,AL |
| E000 89D5 | 475 | MOV AL,16 |
| E000 89D6 | 476 | OUT 0A0H,AL |
| E000 89D7 | 477 | MOV AL,16 |
| E000 89D8 | 478 | OUT 0A0H,AL |
| E000 89D9 | 479 | MOV AL,16 |
| E000 89DA | 480 | OUT 0A0H,AL |
| E000 89DB | 481 | MOV AL,16 |
| E000 89DC | 482 | OUT 0A0H,AL |
| E000 89DD | 483 | MOV AL,16 |
| E000 89DE | 484 | OUT 0A0H,AL |
| E000 89DF | 485 | MOV AL,16 |
| E000 89E0 | 486 | OUT 0A0H,AL |
| E000 89E1 | 487 | MOV AL,16 |
| E000 89E2 | 488 | OUT 0A0H,AL |
| E000 89E3 | 489 | MOV AL,16 |
| E000 89E4 | 490 | OUT 0A0H,AL |
| E000 89E5 | 491 | MOV AL,16 |
| E000 89E6 | 492 | OUT 0A0H,AL |
| E000 89E7 | 493 | MOV AL,16 |
| E000 89E8 | 494 | OUT 0A0H,AL |
| E000 89E9 | 495 | MOV AL,16 |
| E000 89EA | 496 | OUT 0A0H,AL |
| E000 89EB | 497 | MOV AL,16 |
| E000 89EC | 498 | OUT 0A0H,AL |
| E000 89ED | 499 | MOV AL,16 |
| E000 89EE | 500 | OUT 0A0H,AL |
| E000 89EF | 501 | MOV AL,16 |
| E000 89F0 | 502 | OUT 0A0H,AL |
| E000 89F1 | 503 | MOV AL,16 |
| E000 89F2 | 504 | OUT 0A0H,AL |
| E000 89F3 | 505 | MOV AL,16 |
| E000 89F4 | 506 | OUT 0A0H,AL |
| E000 89F5 | 507 | MOV AL,16 |
| E000 89F6 | 508 | OUT 0A0H,AL |
| E000 89F7 | 509 | MOV AL,16 |
| E000 89F8 | 510 | OUT 0A0H,AL |
| E000 89F9 | 511 | MOV AL,16 |
| E000 89FA | 512 | OUT 0A0H,AL |
| E000 89FB | 513 | MOV AL,16 |
| E000 89FC | 514 | OUT 0A0H,AL |
| E000 89FD | 515 | MOV AL,16 |
| E000 89FE | 516 | OUT 0A0H,AL |
| E000 89FF | 517 | MOV AL,16 |
| E000 8A00 | 518 | OUT 0A0H,AL |
| E000 8A01 | 519 | MOV AL,16 |
| E000 8A02 | 520 | OUT 0A0H,AL |
| E000 8A03 | 521 | MOV AL,16 |
| E000 8A04 | 522 | OUT 0A0H,AL |
| E000 8A05 | 523 | MOV AL,16 |
| E000 8A06 | 524 | OUT 0A0H,AL |
| E000 8A07 | 525 | MOV AL,16 |
| E000 8A08 | 526 | OUT 0A0H,AL |
| E000 8A09 | 527 | MOV AL,16 |
| E000 8A0A | 528 | OUT 0A0H,AL |
| E000 8A0B | 529 | MOV AL,16 |
| E000 8A0C | 530 | OUT 0A0H,AL |
| E000 8A0D | 531 | MOV AL,16 |
| E000 8A0E | 532 | OUT 0A0H,AL |
| E000 8A0F | 533 | MOV AL,16 |
| E000 8A10 | 534 | OUT 0A0H,AL |
| E000 8A11 | 535 | MOV AL,16 |
| E000 8A12 | 536 | OUT 0A0H,AL |
| E000 8A13 | 537 | MOV AL,16 |
| E000 8A14 | 538 | OUT 0A0H,AL |
| E000 8A15 | 539 | MOV AL,16 |
| E000 8A16 | 540 | OUT 0A0H,AL |
| E000 8A17 | 541 | MOV AL,16 |
| E000 8A18 | 542 | OUT 0A0H,AL |
| E000 8A19 | 543 | MOV AL,16 |
| E000 8A1A | 544 | OUT 0A0H,AL |
| E000 8A1B | 545 | MOV AL,16 |
| E000 8A1C | 546 | OUT 0A0H,AL |
| E000 8A1D | 547 | MOV AL,16 |
| E000 8A1E | 548 | OUT 0A0H,AL |
| E000 8A1F | 549 | MOV AL,16 |
| E000 8A20 | 550 | OUT 0A0H,AL |
| E000 8A21 | 551 | MOV AL,16 |
| E000 8A22 | 552 | OUT 0A0H,AL |
| E000 8A23 | 553 | MOV AL,16 |
| E000 8A24 | 554 | OUT 0A0H,AL |
| E000 8A25 | 555 | MOV AL,16 |
| E000 8A26 | 556 | OUT 0A0H,AL |
| E000 8A27 | 557 | MOV AL,16 |
| E000 8A28 | 558 | OUT 0A0H,AL |
| E000 8A29 | 559 | MOV AL,16 |
| E000 8A2A | 560 | OUT 0A0H,AL |
| E000 8A2B | 561 | MOV AL,16 |
| E000 8A2C | 562 | OUT 0A0H,AL |
| E000 8A2D | 563 | MOV AL,16 |
| E000 8A2E | 564 | OUT 0A0H,AL |
| E000 8A2F | 565 | MOV AL,16 |
| E000 8A30 | 566 | OUT 0A0H,AL |
| E000 8A31 | 567 | MOV AL,16 |
| E000 8A32 | 568 | OUT 0A0H,AL |
| E000 8A33 | 569 | MOV AL,16 |
| E000 8A34 | 570 | OUT 0A0H,AL |
| E000 8A35 | 571 | MOV AL,16 |
| E000 8A36 | 572 | OUT 0A0H,AL |
| E000 8A37 | 573 | MOV AL,16 |
| E000 8A38 | 574 | OUT 0A0H,AL |
| E000 8A39 | 575 | MOV AL,16 |
| E000 8A3A | 576 | OUT 0A0H,AL |
| E000 8A3B | 577 | MOV AL,16 |
| E000 8A3C | 578 | OUT 0A0H,AL |
| E000 8A3D | 579 | MOV AL,16 |
| E000 8A3E | 580 | OUT 0A0H,AL |
| E000 8A3F | 581 | MOV AL,16 |
| E000 8A40 | 582 | OUT 0A0H,AL |
| E000 8A41 | 583 | MOV AL,16 |
| E000 8A42 | 584 | OUT 0A0H,AL |
| E000 8A43 | 585 | MOV AL,16 |
| E000 8A44 | 586 | OUT 0A0H,AL |
| E000 8A45 | 587 | MOV AL,16 |
| E000 8A46 | 588 | OUT 0A0H,AL |
| E000 8A47 | 589 | MOV AL,16 |
| E000 8A48 | 590 | OUT 0A0H,AL |
| E000 8A49 | 591 | MOV AL,16 |
| E000 8A4A | 592 | OUT 0A0H,AL |
| E000 8A4B | 593 | MOV AL,16 |
| E000 8A4C | 594 | OUT 0A0H,AL |
| E000 8A4D | 595 | MOV AL,16 |
| E000 8A4E | 596 | OUT 0A0H,AL |
| E000 8A4F | 597 | MOV AL,16 |
| E000 8A50 | 598 | OUT 0A0H,AL |
| E000 8A51 | 599 | MOV AL,16 |
| E000 8A52 | 600 | OUT 0A0H,AL |
| E000 8A53 | 601 | MOV AL,16 |
| E000 8A54 | 602 | OUT 0A0H,AL |
| E000 8A55 | 603 | MOV AL,16 |
| E000 8A56 | 604 | OUT 0A0H,AL |
| E000 8A57 | 605 | MOV AL,16 |
| E000 8A58 | 606 | OUT 0A0H,AL |
| E000 8A59 | 607 | MOV AL,16 |
| E000 8A5A | 608 | OUT 0A0H,AL |
| E000 8A5B | 609 | MOV AL,16 |
| E000 8A5C | 610 | OUT 0A0H,AL |
| E000 8A5D | 611 | MOV AL,16 |
| E000 8A5E | 612 | OUT 0A0H,AL |
| E000 8A5F | 613 | MOV AL,16 |
| E000 8A60 | 614 | OUT 0A0H,AL |
| E000 8A61 | 615 | MOV AL,16 |
| E000 8A62 | 616 | OUT 0A0H,AL |
| E000 8A63 | 617 | MOV AL,16 |
| E000 8A64 | 618 | OUT 0A0H,AL |
| E000 8A65 | 619 | MOV AL,16 |
| E000 8A66 | 620 | OUT 0A0H,AL |
| E000 8A67 | 621 | MOV AL,16 |
| E000 8A68 | 622 | OUT 0A0H,AL |
| E000 8A69 | 623 | MOV AL,16 |
| E000 8A6A | 624 | OUT 0A0H,AL |
| E000 8A6B | 625 | MOV AL,16 |
| E000 8A6C | 626 | OUT 0A0H,AL |
| E000 8A6D | 627 | MOV AL,16 |
| E000 8A6E | 628 | OUT 0A0H,AL |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|-----------------|------|------------------------------|------------------------|------|---|
| E179 E2FD | 462 | LOOP C19 | E1FA 2AC706A00C3E2 R | 537 | MOV ES:INT_PTR,OFFSET INT1_INT |
| | 463 | | E201 2AC706A0A0000F0 R | 538 | MOV ES:INT_PTR+2,CODE |
| | 464 | | E208 E92A00 | 539 | JMP TS16 |
| | 465 | | | 540 | ROS_CHECKSUM |
| E17B E462 | 466 | I1 AL,PORT_C | E20B 800020 | 541 | MOV CX,DIY2 |
| E17D 2A0F | 467 | AND AL,0FH | E20C 32C0 | 543 | XOR AL,AL |
| E17F 7A18 | 468 | JZ C21 | E210 | 544 | C26: |
| E181 BA0010 | 469 | MOV DX,1000H | E210 2E0207 | 545 | ADD AL,C5:10X1 |
| E184 8A00 | 470 | MOV AH,AL | E213 43 | 546 | INC BX |
| E186 8000 | 471 | MOV AL,0 | E214 E2FA | 547 | LOOP C26 |
| E188 00C5 | 472 | | E216 0AC0 | 548 | OR AL,AL |
| E18A 80C0 | 473 | MOV ES,DX | E218 C3 | 549 | BET |
| E18C 900000 | 474 | MOV CX,8000H | ROS_CHECKSUM | 550 | ENDP |
| E18D 20FF | 475 | SUB DI,DI | | 551 | ----- |
| E18F F3 | 476 | REP STOSB | | 552 | : INITIAL RELIABILITY TEST -- PHASE 2 |
| | 477 | | | 553 | ----- |
| E190 AA | 478 | ADD DX,8000H | | 554 | ASSUME CS:CODE,ES:ABS0 |
| E191 81C20000 | 479 | DEC AH | D1 DB | 555 | 'PARITY CHECK 2' |
| E195 FECC | 480 | JNZ C20 | D1L EQU | 556 | '-01 |
| E197 75EF | 481 | | D2 DB | 557 | '-PARITY CHECK 1' |
| | 482 | | | 558 | ----- |
| | 483 | | | 559 | D2L EQU |
| E199 8013 | 484 | MOV AL,13H | | 560 | TEST 06 |
| E19B E620 | 485 | OUT INTA00,AL | | 561 | ----- |
| E19D 8008 | 486 | MOV AL,0 | | 562 | : 8259 INTERRUPT CONTROLLER TEST |
| E19F E621 | 487 | OUT INTA01,AL | | 563 | DESCRIPTION |
| E1A1 8009 | 488 | MOV AL,9 | | 564 | : READ/WRITE THE INTERRUPT MASK REGISTER (IMR) WITH ALL ONES AND ZEROS. |
| E1A3 E621 | 489 | OUT INTA01,AL | | 565 | : ENABLE SYSTEM INTERRUPTS. MASK DEVICE INTERRUPTS OFF. CHECK FOR |
| E1A5 28C0 | 490 | SUB AX,AX | | 566 | : NOT INTERRUPTS (UNEXPECTED). |
| E1A7 80C0 | 491 | MOV ES,AX | | 567 | TS16: |
| E1A9 BEA000 | 492 | MOV SI,DATA | | 568 | ----- |
| E1AC 80C0 | 493 | MOV DS,SI | E235 | 569 | SET UP ES REG |
| E1AE 81E72000 | 494 | MOV RESET_FLAG,BX | E235 28C0 | 570 | MOV ES,AX |
| E1B0 01C7200A12 | 495 | CMR RESET_FLAG,1234H | E237 0EC0 | 571 | |
| E1B2 7A18 | 496 | JE C25 | | 572 | ----- |
| E1B4 80C0 | 497 | MOV DS,AX | | 573 | :----- SET UP THE INTERRUPT 5 POINTER TO A DUMMY |
| | 498 | | | 574 | MOV ES:INT5_PTR,OFFSET PRINT_SCREEN |
| | 499 | | | 575 | MOV ES:INT5_PTR+2,CODE |
| | 500 | | | 576 | |
| E1BC BCF03F | 501 | MOV SP,3F0H | | 577 | TEST THE IMR REGISTER |
| E1BF 80C0 | 502 | MOV SS,AX | | 578 | CLI |
| E1C1 80F0 | 503 | MOV DI,AX | | 579 | MOV AL,0 |
| E1C3 B82A00 | 504 | MOV BX,2AH | | 580 | MOV AL,0 |
| E1C5 C70706A2 | 505 | MOV WORD PTR [BX],OFFSET D11 | | 581 | OUT INTA01,AL |
| E1C7 43 | 506 | INC BX | | 582 | IN AL,INTA01 |
| E1C9 AC0F | 507 | MOV [BX],CS | | 583 | OR AL,AL |
| E1CB E8794 | 508 | CALL KBD RESET | | 584 | OR AL,AL |
| E1CD 80F6A5 | 509 | CALL KBD RESET | | 585 | JLZ D6 |
| E1D1 80F6A5 | 510 | CALL BL,0ASH | | 586 | MOV AL,OFFH |
| E1D4 750E | 511 | JNZ C23 | | 587 | MOV AL,OFFH |
| E1D6 B2FF | 512 | CALL SP,TEST | | 588 | OUT INTA01,AL |
| E1D8 F8A04 | 513 | CALL SP,TEST | | 589 | IN AL,INTA01 |
| E1DA 8AC3 | 514 | MOV AL,BL | | 590 | ADD AL,1 |
| E1DD AA | 515 | STOSB | | 591 | JNZ D6 |
| E1DE FECA | 516 | DEC DL | | 592 | |
| E1E0 75F6 | 517 | JNZ C22 | | 593 | CHECK FOR HOT INTERRUPTS |
| E1E2 C03E | 518 | INT 3EH | | 594 | CLD |
| E1E4 0E | 519 | PUSH CS | | 595 | MOV CX,0 |
| E1E5 17 | 520 | POP SS | | 596 | MOV DI,OFFSET INT_PTR |
| E1E6 FA | 521 | CLI | | 597 | D3: |
| E1E7 BC1800 | 522 | MOV SP,OFFSET C2 | | 598 | MOV AX,OFFSET D11 |
| E1EA E320FE | 523 | CALL STGTEST | | 599 | STOSM |
| E1ED 7A03 | 524 | JE C25 | | 600 | MOV AX,CODE |
| E1EF E900FE | 525 | JMP E901 | | 601 | STOSM |
| | 526 | | | 602 | ADD BX,4 |
| | 527 | | | 603 | LOOP D3 |
| | 528 | | | 604 | INTERUPTS ARE MASKED OFF. CHECK THAT NO INTERRUPTS OCCUR. |
| | 529 | | | 605 | MOV AX,AX |
| E1FE | 530 | MOV AX,STACK | | 606 | XOR AX,AX |
| E1FF 803000 | 531 | MOV SS,AX | | 607 | STI |
| E1FF 80001 | 532 | MOV SP,OFFSET TOS | | 608 | SUB CX,CX |
| | 533 | | | 609 | DA: |
| | 534 | | | 610 | LOOP DS |
| | 535 | | | 611 | OR AX,AX |
| | 536 | | | | |

A-10

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|---------|------|---|---------|------|--|
| E352 | 755 | TEST_08 | E300 | 824 | TEST_10 |
| E353 | 756 | INITIALIZE AND START CRT CONTROLLER (68451) | E301 | 825 | GET INTERFACE LINES TEST |
| E354 | 757 | TEST VIDEO READ/WRITE STORAGE. | E302 | 826 | DESCRIPTION |
| E355 | 758 | RESET THE VIDEO ENABLE SIGNAL. | E303 | 827 | SENSE ON/OFF TRANSITION OF THE VIDEO ENABLE AND HORIZONTAL |
| E356 | 759 | SELECT ALPHANUMERIC MODE, 40 * 25, B & W. | E304 | 828 | STHC LINES. |
| E357 | 760 | READ/WRITE DATA PATT. TO STG. CHECK STG ADDRESSABILITY. | E305 | 829 | POP AX |
| E358 | 761 | TEST_09 | E306 | 830 | PUSH AX |
| E359 | 762 | IN AL,PORT_A | E307 | 831 | POP AX |
| E360 | 763 | MOV AH,0 | E308 | 832 | PUSH AX |
| E361 | 764 | MOV AH,0 | E309 | 833 | POP AX |
| E362 | 765 | MOV AH,0 | E310 | 834 | POP AX |
| E363 | 766 | MOV AH,0 | E311 | 835 | POP AX |
| E364 | 767 | MOV AH,0 | E312 | 836 | POP AX |
| E365 | 768 | MOV AH,0 | E313 | 837 | POP AX |
| E366 | 769 | MOV AH,0 | E314 | 838 | POP AX |
| E367 | 770 | MOV AH,0 | E315 | 839 | POP AX |
| E368 | 771 | MOV AH,0 | E316 | 840 | POP AX |
| E369 | 772 | MOV AH,0 | E317 | 841 | POP AX |
| E370 | 773 | MOV AH,0 | E318 | 842 | POP AX |
| E371 | 774 | MOV AH,0 | E319 | 843 | POP AX |
| E372 | 775 | MOV AH,0 | E320 | 844 | POP AX |
| E373 | 776 | MOV AH,0 | E321 | 845 | POP AX |
| E374 | 777 | MOV AH,0 | E322 | 846 | POP AX |
| E375 | 778 | MOV AH,0 | E323 | 847 | POP AX |
| E376 | 779 | MOV AH,0 | E324 | 848 | POP AX |
| E377 | 780 | MOV AH,0 | E325 | 849 | POP AX |
| E378 | 781 | MOV AH,0 | E326 | 850 | POP AX |
| E379 | 782 | MOV AH,0 | E327 | 851 | POP AX |
| E380 | 783 | MOV AH,0 | E328 | 852 | POP AX |
| E381 | 784 | MOV AH,0 | E329 | 853 | POP AX |
| E382 | 785 | MOV AH,0 | E330 | 854 | POP AX |
| E383 | 786 | MOV AH,0 | E331 | 855 | POP AX |
| E384 | 787 | MOV AH,0 | E332 | 856 | POP AX |
| E385 | 788 | MOV AH,0 | E333 | 857 | POP AX |
| E386 | 789 | MOV AH,0 | E334 | 858 | POP AX |
| E387 | 790 | MOV AH,0 | E335 | 859 | POP AX |
| E388 | 791 | MOV AH,0 | E336 | 860 | POP AX |
| E389 | 792 | MOV AH,0 | E337 | 861 | POP AX |
| E390 | 793 | MOV AH,0 | E338 | 862 | POP AX |
| E391 | 794 | MOV AH,0 | E339 | 863 | POP AX |
| E392 | 795 | MOV AH,0 | E340 | 864 | POP AX |
| E393 | 796 | MOV AH,0 | E341 | 865 | POP AX |
| E394 | 797 | MOV AH,0 | E342 | 866 | POP AX |
| E395 | 798 | MOV AH,0 | E343 | 867 | POP AX |
| E396 | 799 | MOV AH,0 | E344 | 868 | POP AX |
| E397 | 800 | MOV AH,0 | E345 | 869 | POP AX |
| E398 | 801 | MOV AH,0 | E346 | 870 | POP AX |
| E399 | 802 | MOV AH,0 | E347 | 871 | POP AX |
| E400 | 803 | MOV AH,0 | E348 | 872 | POP AX |
| E401 | 804 | MOV AH,0 | E349 | 873 | POP AX |
| E402 | 805 | MOV AH,0 | E350 | 874 | POP AX |
| E403 | 806 | MOV AH,0 | E351 | 875 | POP AX |
| E404 | 807 | MOV AH,0 | E352 | 876 | POP AX |
| E405 | 808 | MOV AH,0 | E353 | 877 | POP AX |
| E406 | 809 | MOV AH,0 | E354 | 878 | POP AX |
| E407 | 810 | MOV AH,0 | E355 | 879 | POP AX |
| E408 | 811 | MOV AH,0 | E356 | 880 | POP AX |
| E409 | 812 | MOV AH,0 | E357 | 881 | POP AX |
| E410 | 813 | MOV AH,0 | E358 | 882 | POP AX |
| E411 | 814 | MOV AH,0 | E359 | 883 | POP AX |
| E412 | 815 | MOV AH,0 | E360 | 884 | POP AX |
| E413 | 816 | MOV AH,0 | E361 | 885 | POP AX |
| E414 | 817 | MOV AH,0 | E362 | 886 | POP AX |
| E415 | 818 | MOV AH,0 | E363 | 887 | POP AX |
| E416 | 819 | MOV AH,0 | E364 | 888 | POP AX |
| E417 | 820 | MOV AH,0 | E365 | 889 | POP AX |
| E418 | 821 | MOV AH,0 | E366 | 890 | POP AX |
| E419 | 822 | MOV AH,0 | E367 | 891 | POP AX |
| E420 | 823 | MOV AH,0 | E368 | 892 | POP AX |
| E421 | 824 | MOV AH,0 | E369 | 893 | POP AX |
| E422 | 825 | MOV AH,0 | E370 | 894 | POP AX |
| E423 | 826 | MOV AH,0 | E371 | 895 | POP AX |
| E424 | 827 | MOV AH,0 | E372 | 896 | POP AX |
| E425 | 828 | MOV AH,0 | E373 | 897 | POP AX |
| E426 | 829 | MOV AH,0 | E374 | 898 | POP AX |
| E427 | 830 | MOV AH,0 | E375 | 899 | POP AX |
| E428 | 831 | MOV AH,0 | E376 | 900 | POP AX |
| E429 | 832 | MOV AH,0 | E377 | 901 | POP AX |
| E430 | 833 | MOV AH,0 | E378 | 902 | POP AX |
| E431 | 834 | MOV AH,0 | E379 | 903 | POP AX |
| E432 | 835 | MOV AH,0 | E380 | 904 | POP AX |
| E433 | 836 | MOV AH,0 | E381 | 905 | POP AX |
| E434 | 837 | MOV AH,0 | E382 | 906 | POP AX |
| E435 | 838 | MOV AH,0 | E383 | 907 | POP AX |
| E436 | 839 | MOV AH,0 | E384 | 908 | POP AX |
| E437 | 840 | MOV AH,0 | E385 | 909 | POP AX |
| E438 | 841 | MOV AH,0 | E386 | 910 | POP AX |
| E439 | 842 | MOV AH,0 | E387 | 911 | POP AX |
| E440 | 843 | MOV AH,0 | E388 | 912 | POP AX |
| E441 | 844 | MOV AH,0 | E389 | 913 | POP AX |
| E442 | 845 | MOV AH,0 | E390 | 914 | POP AX |
| E443 | 846 | MOV AH,0 | E391 | 915 | POP AX |
| E444 | 847 | MOV AH,0 | E392 | 916 | POP AX |
| E445 | 848 | MOV AH,0 | E393 | 917 | POP AX |
| E446 | 849 | MOV AH,0 | E394 | 918 | POP AX |
| E447 | 850 | MOV AH,0 | E395 | 919 | POP AX |
| E448 | 851 | MOV AH,0 | E396 | 920 | POP AX |
| E449 | 852 | MOV AH,0 | E397 | 921 | POP AX |
| E450 | 853 | MOV AH,0 | E398 | 922 | POP AX |
| E451 | 854 | MOV AH,0 | E399 | 923 | POP AX |
| E452 | 855 | MOV AH,0 | E400 | 924 | POP AX |
| E453 | 856 | MOV AH,0 | E401 | 925 | POP AX |
| E454 | 857 | MOV AH,0 | E402 | 926 | POP AX |
| E455 | 858 | MOV AH,0 | E403 | 927 | POP AX |
| E456 | 859 | MOV AH,0 | E404 | 928 | POP AX |
| E457 | 860 | MOV AH,0 | E405 | 929 | POP AX |
| E458 | 861 | MOV AH,0 | E406 | 930 | POP AX |
| E459 | 862 | MOV AH,0 | E407 | 931 | POP AX |
| E460 | 863 | MOV AH,0 | E408 | 932 | POP AX |
| E461 | 864 | MOV AH,0 | E409 | 933 | POP AX |
| E462 | 865 | MOV AH,0 | E410 | 934 | POP AX |
| E463 | 866 | MOV AH,0 | E411 | 935 | POP AX |
| E464 | 867 | MOV AH,0 | E412 | 936 | POP AX |
| E465 | 868 | MOV AH,0 | E413 | 937 | POP AX |
| E466 | 869 | MOV AH,0 | E414 | 938 | POP AX |
| E467 | 870 | MOV AH,0 | E415 | 939 | POP AX |
| E468 | 871 | MOV AH,0 | E416 | 940 | POP AX |
| E469 | 872 | MOV AH,0 | E417 | 941 | POP AX |
| E470 | 873 | MOV AH,0 | E418 | 942 | POP AX |
| E471 | 874 | MOV AH,0 | E419 | 943 | POP AX |
| E472 | 875 | MOV AH,0 | E420 | 944 | POP AX |
| E473 | 876 | MOV AH,0 | E421 | 945 | POP AX |
| E474 | 877 | MOV AH,0 | E422 | 946 | POP AX |
| E475 | 878 | MOV AH,0 | E423 | 947 | POP AX |
| E476 | 879 | MOV AH,0 | E424 | 948 | POP AX |
| E477 | 880 | MOV AH,0 | E425 | 949 | POP AX |
| E478 | 881 | MOV AH,0 | E426 | 950 | POP AX |
| E479 | 882 | MOV AH,0 | E427 | 951 | POP AX |
| E480 | 883 | MOV AH,0 | E428 | 952 | POP AX |
| E481 | 884 | MOV AH,0 | E429 | 953 | POP AX |
| E482 | 885 | MOV AH,0 | E430 | 954 | POP AX |
| E483 | 886 | MOV AH,0 | E431 | 955 | POP AX |
| E484 | 887 | MOV AH,0 | E432 | 956 | POP AX |
| E485 | 888 | MOV AH,0 | E433 | 957 | POP AX |
| E486 | 889 | MOV AH,0 | E434 | 958 | POP AX |
| E487 | 890 | MOV AH,0 | E435 | 959 | POP AX |
| E488 | 891 | MOV AH,0 | E436 | 960 | POP AX |
| E489 | 892 | MOV AH,0 | E437 | 961 | POP AX |
| E490 | 893 | MOV AH,0 | E438 | 962 | POP AX |
| E491 | 894 | MOV AH,0 | E439 | 963 | POP AX |
| E492 | 895 | MOV AH,0 | E440 | 964 | POP AX |
| E493 | 896 | MOV AH,0 | E441 | 965 | POP AX |
| E494 | 897 | MOV AH,0 | E442 | 966 | POP AX |
| E495 | 898 | MOV AH,0 | E443 | 967 | POP AX |
| E496 | 899 | MOV AH,0 | E444 | 968 | POP AX |
| E497 | 900 | MOV AH,0 | E445 | 969 | POP AX |
| E498 | 901 | MOV AH,0 | E446 | 970 | POP AX |
| E499 | 902 | MOV AH,0 | E447 | 971 | POP AX |
| E500 | 903 | MOV AH,0 | E448 | 972 | POP AX |
| E501 | 904 | MOV AH,0 | E449 | 973 | POP AX |
| E502 | 905 | MOV AH,0 | E450 | 974 | POP AX |
| E503 | 906 | MOV AH,0 | E451 | 975 | POP AX |
| E504 | 907 | MOV AH,0 | E452 | 976 | POP AX |
| E505 | 908 | MOV AH,0 | E453 | 977 | POP AX |
| E506 | 909 | MOV AH,0 | E454 | 978 | POP AX |
| E507 | 910 | MOV AH,0 | E455 | 979 | POP AX |
| E508 | 911 | MOV AH,0 | E456 | 980 | POP AX |
| E509 | 912 | MOV AH,0 | E457 | 981 | POP AX |
| E510 | 913 | MOV AH,0 | E458 | 982 | POP AX |
| E511 | 914 | MOV AH,0 | E459 | 983 | POP AX |
| E512 | 915 | MOV AH,0 | E460 | 984 | POP AX |
| E513 | 916 | MOV AH,0 | E461 | 985 | POP AX |
| E514 | 917 | MOV AH,0 | E462 | 986 | POP AX |
| E515 | 918 | MOV AH,0 | E463 | 987 | POP AX |
| E516 | 919 | MOV AH,0 | E464 | 988 | POP AX |
| E517 | 920 | MOV AH,0 | E465 | 989 | POP AX |
| E518 | 921 | MOV AH,0 | E466 | 990 | POP AX |
| E519 | 922 | MOV AH,0 | E467 | 991 | POP AX |
| E520 | 923 | MOV AH,0 | E468 | 992 | POP AX |
| E521 | 924 | MOV AH,0 | E469 | 993 | POP AX |
| E522 | 925 | MOV AH,0 | E470 | 994 | POP AX |
| E523 | 926 | MOV AH,0 | E471 | 995 | POP AX |
| E524 | 927 | MOV AH,0 | E472 | 996 | POP AX |
| E525 | 928 | MOV AH,0 | E473 | 997 | POP AX |
| E526 | 929 | MOV AH,0 | E474 | 998 | POP AX |
| E527 | 930 | MOV AH,0 | E475 | 999 | POP AX |
| E528 | 931 | MOV AH,0 | E476 | 1000 | POP AX |
| E529 | 932 | MOV AH,0 | E477 | 1001 | POP AX |
| E530 | 933 | MOV AH,0 | E478 | 1002 | POP AX |
| E531 | 934 | MOV AH,0 | E479 | 1003 | POP AX |
| E532 | 935 | MOV AH,0 | E480 | 1004 | POP AX |
| E533 | 936 | MOV AH,0 | E481 | 1005 | POP AX |
| E534 | 937 | MOV AH,0 | E482 | 1006 | POP AX |
| E535 | 938 | MOV AH,0 | E483 | 1007 | POP AX |
| E536 | 939 | MOV AH,0 | E484 | 1008 | POP AX |
| E537 | 940 | MOV AH,0 | E485 | 1009 | POP AX |
| E538 | 941 | MOV AH,0 | E486 | 1010 | POP AX |
| E539 | 942 | MOV AH,0 | E487 | 1011 | POP AX |
| E540 | 943 | MOV AH,0 | E488 | 1012 | POP AX |
| E541 | 944 | MOV AH,0 | E489 | 1013 | POP AX |
| E542 | 945 | MOV AH,0 | E490 | 1014 | POP AX |
| E543 | 946 | MOV AH,0 | E491 | 1015 | POP AX |
| E544 | 947 | MOV AH,0 | E492 | 1016 | POP AX |
| E545 | 948 | MOV AH,0 | E493 | 1017 | POP AX |
| E546 | 949 | MOV AH,0 | E494 | 1018 | POP AX |
| E547 | 950 | MOV AH,0 | E495 | 1019 | POP AX |
| E548 | 951 | MOV AH,0 | E496 | 1020 | POP AX |
| E549 | 952 | MOV AH,0 | E497 | 1021 | POP AX |
| E550 | 953 | MOV AH,0 | E498 | 1022 | POP AX |
| E551 | 954 | MOV AH,0 | E499 | 1023 | POP AX |
| E552 | 955 | MOV AH,0 | E500 | 1024 | POP AX |
| E553 | 956 | MOV AH,0 | E501 | 1025 | POP AX |
| E554 | 957 | MOV AH,0 | E502 | 1026 | POP AX |
| E555 | 958 | MOV AH,0 | E503 | 1027 | POP AX |
| E556 | 959 | MOV AH,0 | E504 | 1028 | POP AX |
| E557 | 960 | MOV AH,0 | E505 | 1029 | POP AX |
| E558 | 961 | MOV AH,0 | E506 | 1030 | POP AX |
| E559 | 962 | MOV AH,0 | E507 | 1031 | POP AX |
| E560 | 963 | MOV AH,0 | E508 | 1032 | POP AX |
| E561 | 964 | MOV AH,0 | E509 | 1033 | POP AX |
| E562 | 965 | MOV AH,0 | E510 | 1034 | POP AX |
| E563 | 966 | MOV AH,0 | E511 | 1035 | POP AX |
| E564 | 967 | MOV AH,0 | E512 | 1036 | POP AX |
| E565 | 968 | MOV AH,0 | E513 | 1037 | POP AX |
| E566 | 969 | MOV AH,0 | E514 | 1038 | POP AX |
| E567 | 970 | MOV AH,0 | E515 | 1039 | POP AX |
| E568 | 971 | MOV AH,0 | E516 | 1040 | POP AX |
| E569 | 972 | MOV AH,0 | E517 | 1041 | POP AX |
| E570 | 973 | MOV AH,0 | E518 | 1042 | POP AX |
| E571 | 974 | MOV AH,0 | E519 | 1043 | POP AX |
| E572 | 975 | MOV AH,0 | E520 | 1044 | POP AX |
| E573 | 976 | MOV AH,0 | E521 | 1045 | POP AX |
| E574 | 977 | MOV AH,0 | E522 | 1046 | POP AX |
| E575 | 978 | MOV AH,0 | E523 | 1047 | POP AX |
| E576 | 979 | MOV AH,0 | E524 | 1048 | POP AX |
| E577 | 980 | | | | |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|-------------|------|--|-------------|------|--|
| E428 7440 | 900 | JE E22 | E428 7440 | 900 | JE E22 |
| E429 8A0004 | 901 | TEST ANY OTHER READ/WRITE STORAGE AVAILABLE | E429 8A0004 | 901 | TEST ANY OTHER READ/WRITE STORAGE AVAILABLE |
| E430 891000 | 902 | | E430 891000 | 902 | |
| E431 | 903 | | E431 | 903 | |
| E432 3001 | 904 | MOV BX,400H | E432 3001 | 904 | MOV BX,400H |
| E433 7846 | 905 | MOV CX,16 | E433 7846 | 905 | MOV CX,16 |
| E434 8000 | 906 | | E434 8000 | 906 | |
| E435 8000 | 907 | CHP DX,CX | E435 8000 | 907 | CHP DX,CX |
| E436 8000 | 908 | JE E23 | E436 8000 | 908 | JE E23 |
| E437 8000 | 909 | MOV ES,BX | E437 8000 | 909 | MOV ES,BX |
| E438 8000 | 910 | MOV ES,BX | E438 8000 | 910 | MOV ES,BX |
| E439 8000 | 911 | ADD CX,16 | E439 8000 | 911 | ADD CX,16 |
| E440 8000 | 912 | ADD BX,400H | E440 8000 | 912 | ADD BX,400H |
| E441 8000 | 913 | PUSH CX | E441 8000 | 913 | PUSH CX |
| E442 51 | 914 | PUSH BX | E442 51 | 914 | PUSH BX |
| E443 53 | 915 | PUSH DX | E443 53 | 915 | PUSH DX |
| E444 52 | 916 | CALL STGTST | E444 52 | 916 | CALL STGTST |
| E445 8000 | 917 | POP DX | E445 8000 | 917 | POP DX |
| E446 5A | 918 | POP BX | E446 5A | 918 | POP BX |
| E447 5B | 919 | POP CX | E447 5B | 919 | POP CX |
| E448 59 | 920 | JE E21 | E448 59 | 920 | JE E21 |
| E449 74E6 | 921 | PRINT FAILING ADDRESS AND XOR'ED PATTERN IF DATA COMPARE ERROR | E449 74E6 | 921 | PRINT FAILING ADDRESS AND XOR'ED PATTERN IF DATA COMPARE ERROR |
| E450 8000 | 922 | MOV DX,05 | E450 8000 | 922 | MOV DX,05 |
| E451 8000 | 923 | MOV CH,AL | E451 8000 | 923 | MOV CH,AL |
| E452 8000 | 924 | MOV AL,CH | E452 8000 | 924 | MOV AL,CH |
| E453 8000 | 925 | MOV CL,A | E453 8000 | 925 | MOV CL,A |
| E454 8000 | 926 | SHR AL,CL | E454 8000 | 926 | SHR AL,CL |
| E455 8000 | 927 | CALL XLAT_PRINT_CODE | E455 8000 | 927 | CALL XLAT_PRINT_CODE |
| E456 8000 | 928 | AND AL,0FH | E456 8000 | 928 | AND AL,0FH |
| E457 8000 | 929 | CALL XLAT_PRINT_CODE | E457 8000 | 929 | CALL XLAT_PRINT_CODE |
| E458 8000 | 930 | MOV CL,A | E458 8000 | 930 | MOV CL,A |
| E459 8000 | 931 | SHR AL,CL | E459 8000 | 931 | SHR AL,CL |
| E460 8000 | 932 | CALL XLAT_PRINT_CODE | E460 8000 | 932 | CALL XLAT_PRINT_CODE |
| E461 8000 | 933 | MOV CL,A | E461 8000 | 933 | MOV CL,A |
| E462 8000 | 934 | SHR AL,CL | E462 8000 | 934 | SHR AL,CL |
| E463 8000 | 935 | CALL XLAT_PRINT_CODE | E463 8000 | 935 | CALL XLAT_PRINT_CODE |
| E464 8000 | 936 | MOV AL,CH | E464 8000 | 936 | MOV AL,CH |
| E465 8000 | 937 | AND AL,0FH | E465 8000 | 937 | AND AL,0FH |
| E466 8000 | 938 | CALL XLAT_PRINT_CODE | E466 8000 | 938 | CALL XLAT_PRINT_CODE |
| E467 8000 | 939 | MOV SI,OFFSET E1 | E467 8000 | 939 | MOV SI,OFFSET E1 |
| E468 8000 | 940 | MOV CX,E1 | E468 8000 | 940 | MOV CX,E1 |
| E469 8000 | 941 | CALL P_MSG | E469 8000 | 941 | CALL P_MSG |
| E470 8000 | 942 | CALL P_MSG | E470 8000 | 942 | CALL P_MSG |
| E471 8000 | 943 | JMP TEST12 | E471 8000 | 943 | JMP TEST12 |
| E472 8000 | 944 | JMP TEST12 | E472 8000 | 944 | JMP TEST12 |
| E473 8000 | 945 | MOV AX,DATA | E473 8000 | 945 | MOV AX,DATA |
| E474 8000 | 946 | MOV DS,AX | E474 8000 | 946 | MOV DS,AX |
| E475 8000 | 947 | MOV DX,05 | E475 8000 | 947 | MOV DX,05 |
| E476 8000 | 948 | MOV DX,05 | E476 8000 | 948 | MOV DX,05 |
| E477 8000 | 949 | MOV DX,05 | E477 8000 | 949 | MOV DX,05 |
| E478 8000 | 950 | MOV DX,05 | E478 8000 | 950 | MOV DX,05 |
| E479 8000 | 951 | MOV DX,05 | E479 8000 | 951 | MOV DX,05 |
| E480 8000 | 952 | MOV DX,05 | E480 8000 | 952 | MOV DX,05 |
| E481 8000 | 953 | MOV DX,05 | E481 8000 | 953 | MOV DX,05 |
| E482 8000 | 954 | MOV DX,05 | E482 8000 | 954 | MOV DX,05 |
| E483 8000 | 955 | MOV DX,05 | E483 8000 | 955 | MOV DX,05 |
| E484 8000 | 956 | MOV DX,05 | E484 8000 | 956 | MOV DX,05 |
| E485 8000 | 957 | MOV DX,05 | E485 8000 | 957 | MOV DX,05 |
| E486 8000 | 958 | MOV DX,05 | E486 8000 | 958 | MOV DX,05 |
| E487 8000 | 959 | MOV DX,05 | E487 8000 | 959 | MOV DX,05 |
| E488 8000 | 960 | MOV DX,05 | E488 8000 | 960 | MOV DX,05 |
| E489 8000 | 961 | MOV DX,05 | E489 8000 | 961 | MOV DX,05 |
| E490 8000 | 962 | MOV DX,05 | E490 8000 | 962 | MOV DX,05 |
| E491 8000 | 963 | MOV DX,05 | E491 8000 | 963 | MOV DX,05 |
| E492 8000 | 964 | MOV DX,05 | E492 8000 | 964 | MOV DX,05 |
| E493 8000 | 965 | MOV DX,05 | E493 8000 | 965 | MOV DX,05 |
| E494 8000 | 966 | MOV DX,05 | E494 8000 | 966 | MOV DX,05 |
| E495 8000 | 967 | MOV DX,05 | E495 8000 | 967 | MOV DX,05 |
| E496 8000 | 968 | MOV DX,05 | E496 8000 | 968 | MOV DX,05 |
| E497 8000 | 969 | MOV DX,05 | E497 8000 | 969 | MOV DX,05 |
| E498 8000 | 970 | MOV DX,05 | E498 8000 | 970 | MOV DX,05 |
| E499 8000 | 971 | MOV DX,05 | E499 8000 | 971 | MOV DX,05 |
| E500 8000 | 972 | MOV DX,05 | E500 8000 | 972 | MOV DX,05 |
| E501 8000 | 973 | MOV DX,05 | E501 8000 | 973 | MOV DX,05 |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|-------------|------|---------|---------------|------|--------------------|
| | 1045 | TEST 13 | 5505 B501 | 1122 | MOV CH,1 |
| | 1046 | ; | 5507 80163E00 | 1123 | MOV SECK_STATUS,DL |
| | 1047 | ; | 5508 60F30B | 1124 | CALL SEEK |
| | 1048 | ; | 550E 7207 | 1125 | JC F13 |
| | 1049 | ; | 5510 8522 | 1126 | MOV CH,34 |
| | 1050 | ; | 5512 80E008 | 1127 | CALL SEEK |
| | 1051 | ; | 5515 7309 | 1128 | JNC F14 |
| | 1052 | ; | 5517 | 1129 | F13: |
| | 1053 | ; | 5517 BEAEE4 | 1130 | MOV SI,OFFSET F3 |
| | 1054 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES1E B84000 | 1055 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES21 8D3D | 1056 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES23 B040 | 1057 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES25 E661 | 1058 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1059 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1060 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1061 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1062 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES27 B0FF | 1063 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES29 E621 | 1064 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES2B B086 | 1065 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES2D E643 | 1066 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES2F B80304 | 1067 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES32 E642 | 1068 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES34 8AC4 | 1069 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES36 E642 | 1070 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1071 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1072 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES38 E642 | 1073 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES3A 2410 | 1074 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES3C A24800 | 1075 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES3E E82E14 | 1076 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES42 E83814 | 1077 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES43 E33C | 1078 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES45 7503 | 1079 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES47 E70950 | 1080 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES4A B0BC | 1081 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES4C E621 | 1082 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES4E B400 | 1083 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES50 C013 | 1084 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES52 F6C4FF | 1085 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES55 7520 | 1086 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1087 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1088 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1089 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1090 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1091 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1092 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1093 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1094 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1095 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1096 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES5C B0FC | 1097 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES5E 6231 | 1098 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES60 A01000 | 1099 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES63 4801 | 1100 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES65 7503 | 1101 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES67 E70950 | 1102 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES6A | 1103 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES6C B0BC | 1104 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES6E 6231 | 1105 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES6E B400 | 1106 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES70 C013 | 1107 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES72 F6C4FF | 1108 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES75 7520 | 1109 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1110 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1111 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1112 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES77 BAF203 | 1113 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES7A B01C | 1114 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES7C EE | 1115 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES7D 28C9 | 1116 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES7F | 1117 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES7F E2FE | 1118 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES81 | 1119 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES81 E2FE | 1120 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| ES83 3302 | 1121 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1122 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1123 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1124 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1125 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1126 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1127 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1128 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1129 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1130 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1131 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1132 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1133 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1134 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1135 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1136 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1137 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1138 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1139 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1140 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1141 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1142 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1143 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1144 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1145 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1146 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1147 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1148 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1149 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1150 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1151 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1152 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1153 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1154 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1155 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1156 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1157 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1158 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1159 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1160 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1161 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1162 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1163 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1164 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1165 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1166 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1167 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1168 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1169 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1170 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1171 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1172 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1173 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1174 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1175 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1176 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1177 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1178 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1179 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1180 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1181 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1182 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1183 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1184 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1185 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1186 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1187 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1188 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1189 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1190 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1191 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1192 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1193 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1194 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1195 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1196 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1197 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1198 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1199 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1200 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1201 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1202 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1203 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1204 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1205 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1206 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1207 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1208 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1209 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1210 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1211 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1212 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1213 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1214 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1215 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1216 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1217 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1218 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1219 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1220 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |
| | 1221 | ; | 5519 895380 | 1131 | MOV CX,F3*16 |

| LOC OBJ | LINE | SOURCE |
|-----------------|------|---|
| 6613 803E120001 | 1199 | R |
| 6610 7406 | 1200 | IFG_TST1.1 |
| 661A B40100 | 1201 | JE F21 |
| 6610 B40100 | 1202 | MOV DX:1 |
| 6620 E60100 | 1203 | CALL ERP_BEOP |
| 6620 E60100 | 1204 | F21: JHP BOOT_STRAP |
| 6623 803E120001 | 1205 | R |
| 6623 803E120001 | 1206 | IFG_TST1.1 |
| 6620 7503 | 1207 | JNE F23 |
| 662A E7E7FA | 1208 | JMP START |
| 662D | 1209 | F23: JHP |
| 662D E7E7FF | 1210 | F15 |
| 662D E7E7FF | 1211 | ----- |
| 662D E7E7FF | 1212 | INITIAL RELIABILITY TEST -- SUBROUTINES |
| 662D E7E7FF | 1213 | ASSUME CS:CODE,DS:DATA |
| 662D E7E7FF | 1214 | ----- |
| 662D E7E7FF | 1215 | SUBROUTINES FOR POWER ON DIAGNOSTICS |
| 662D E7E7FF | 1216 | ----- |
| 662D E7E7FF | 1217 | THIS PROCEDURE WILL ISSUE ONE LONG TONE (1.3 SECS) AND ONE OR |
| 662D E7E7FF | 1218 | MORE SHORT TONES (1 SEC) TO INDICATE A FAILURE ON THE PLUNAR |
| 662D E7E7FF | 1219 | BOARD. A BAD RAM MODULE, OR A PROBLEM WITH THE COT. |
| 662D E7E7FF | 1220 | ENTRY PARAMETERS: |
| 662D E7E7FF | 1221 | DI = NUMBER OF LONG TONES TO BEEP |
| 662D E7E7FF | 1222 | DI = NUMBER OF SHORT TONES TO BEEP |
| 662D E7E7FF | 1223 | ----- |
| 662D E7E7FF | 1224 | ERP_BEOP PROC NEAR |
| 662D E7E7FF | 1225 | ----- |
| 662D E7E7FF | 1226 | CALL |
| 662D E7E7FF | 1227 | PUSH DS |
| 662D E7E7FF | 1228 | MOV AX:DATA |
| 662D E7E7FF | 1229 | MOV DS:AX |
| 662D E7E7FF | 1230 | OR DI,DI |
| 662D E7E7FF | 1231 | JZ G3 |
| 662D E7E7FF | 1232 | G1: NO, DO THE SHORT TONES |
| 662D E7E7FF | 1233 | LONG BEOP: |
| 662D E7E7FF | 1234 | DI = NUMBER OF LONG TONES TO BEEP |
| 662D E7E7FF | 1235 | CALL BEOP |
| 662D E7E7FF | 1236 | LOOP G2 |
| 662D E7E7FF | 1237 | DEC DI |
| 662D E7E7FF | 1238 | JLZ G1 |
| 662D E7E7FF | 1239 | IFG_TST1.1 |
| 662D E7E7FF | 1240 | JNE G3 |
| 662D E7E7FF | 1241 | MOV AL:ODH |
| 662D E7E7FF | 1242 | OUT PORT_B,AL |
| 662D E7E7FF | 1243 | JMP SHORT G1 |
| 662D E7E7FF | 1244 | G3: SHORT BEOP: |
| 662D E7E7FF | 1245 | DI = NUMBER OF SHORT TONES TO BEEP |
| 662D E7E7FF | 1246 | CALL BEOP |
| 662D E7E7FF | 1247 | LOOP G4 |
| 662D E7E7FF | 1248 | DEC DI |
| 662D E7E7FF | 1249 | JLZ G3 |
| 662D E7E7FF | 1250 | DO SOME MORE |
| 662D E7E7FF | 1251 | LONG DELAY BEFORE RETURN |
| 662D E7E7FF | 1252 | LOOP G4 |
| 662D E7E7FF | 1253 | POP DS |
| 662D E7E7FF | 1254 | RESTORE ORIG CONTENTS OF DS |
| 662D E7E7FF | 1255 | RETURN TO CALLER |
| 662D E7E7FF | 1256 | ERP_BEOP PROC NEAR |
| 662D E7E7FF | 1257 | ROUTINE TO SOUND BEEPER |
| 662D E7E7FF | 1258 | ----- |
| 662D E7E7FF | 1259 | PROC NEAR |
| 662D E7E7FF | 1260 | MOV AL:0101010B |
| 662D E7E7FF | 1261 | OUT TMR+3,AL |
| 662D E7E7FF | 1262 | MOV AX:533H |
| 662D E7E7FF | 1263 | OUT TMR+2,AL |
| 662D E7E7FF | 1264 | MOV AL:AH |
| 662D E7E7FF | 1265 | OUT TMR+1,AL |
| 662D E7E7FF | 1266 | MOV AL:PORT_B |
| 662D E7E7FF | 1267 | OUT PORT_B,AL |
| 662D E7E7FF | 1268 | OR AL,03 |
| 662D E7E7FF | 1269 | OUT PORT_B,AL |
| 662D E7E7FF | 1270 | SUB CX,CX |
| 662D E7E7FF | 1271 | LOOP G7 |
| 662D E7E7FF | 1272 | DEC BL |
| 662D E7E7FF | 1273 | JLZ G7 |
| 662D E7E7FF | 1274 | MOV AL:AH |
| 662D E7E7FF | 1275 | OUT PORT_B,AL |
| 662D E7E7FF | 1276 | RET |
| 662D E7E7FF | 1277 | BEOP |
| 662D E7E7FF | 1278 | ----- |
| 662D E7E7FF | 1279 | THIS PROCEDURE WILL SEND A SOFTWARE RESET TO THE KEYBOARD. |
| 662D E7E7FF | 1280 | SCAN CODE AX SHOULD BE RETURNED TO THE CPU. |
| 662D E7E7FF | 1281 | ----- |
| 662D E7E7FF | 1282 | KBD_RESET PROC NEAR |
| 662D E7E7FF | 1283 | MOV AL:0CH |
| 662D E7E7FF | 1284 | OUT PORT_B,AL |
| 662D E7E7FF | 1285 | MOV CX:10B82 |
| 662D E7E7FF | 1286 | OR |
| 662D E7E7FF | 1287 | MOV AL:0CH |
| 662D E7E7FF | 1288 | OUT PORT_B,AL |
| 662D E7E7FF | 1289 | SP_TEST: ENTRY FOR MANUFACTURING TEST 2 |
| 662D E7E7FF | 1290 | MOV AL:0CH |
| 662D E7E7FF | 1291 | OUT PORT_B,AL |
| 662D E7E7FF | 1292 | MOV AL:0FH |
| 662D E7E7FF | 1293 | OUT INTA0,AL |
| 662D E7E7FF | 1294 | STI |
| 662D E7E7FF | 1295 | MOV AH:0 |
| 662D E7E7FF | 1296 | SUB CX,CX |
| 662D E7E7FF | 1297 | TEST AH,OFFH |
| 662D E7E7FF | 1298 | JLZ G10 |
| 662D E7E7FF | 1299 | LOOP G9 |
| 662D E7E7FF | 1300 | IN AL,PORT_A |
| 662D E7E7FF | 1301 | MOV BL,AL |
| 662D E7E7FF | 1302 | MOV AL:0CH |
| 662D E7E7FF | 1303 | OUT PORT_B,AL |
| 662D E7E7FF | 1304 | RET |
| 662D E7E7FF | 1305 | KBD_RESET PROC NEAR |
| 662D E7E7FF | 1306 | ----- |
| 662D E7E7FF | 1307 | BLINK LED PROCEDURE FOR HPG BURN-IN AND RUN-IN TESTS |
| 662D E7E7FF | 1308 | LED WILL BLINK APPROXIMATELY .25 SECOND |
| 662D E7E7FF | 1309 | ----- |
| 662D E7E7FF | 1310 | BLINK_INT PROC NEAR |
| 662D E7E7FF | 1311 | STI |
| 662D E7E7FF | 1312 | PUSH CX |
| 662D E7E7FF | 1313 | PUSH AX |
| 662D E7E7FF | 1314 | IN AL,PORT_B |
| 662D E7E7FF | 1315 | AND AL,PORT_B |
| 662D E7E7FF | 1316 | OUT PORT_B,AL |
| 662D E7E7FF | 1317 | SUB CX,CX |
| 662D E7E7FF | 1318 | LOOP B11 |
| 662D E7E7FF | 1319 | OR AL,0FH |
| 662D E7E7FF | 1320 | OUT PORT_B,AL |
| 662D E7E7FF | 1321 | MOV AL:EDI |
| 662D E7E7FF | 1322 | OUT INTA0,AL |
| 662D E7E7FF | 1323 | POP AX |
| 662D E7E7FF | 1324 | POP CX |
| 662D E7E7FF | 1325 | IRET |
| 662D E7E7FF | 1326 | BLINK_INT PROC NEAR |
| 662D E7E7FF | 1327 | ----- |
| 662D E7E7FF | 1328 | THIS SUBROUTINE WILL PRINT A MESSAGE ON THE DISPLAY |
| 662D E7E7FF | 1329 | ----- |
| 662D E7E7FF | 1330 | PRINT MSG PROC NEAR |
| 662D E7E7FF | 1331 | SI = OFFSET(ADDRESS) OF MESSAGE BUFFER |
| 662D E7E7FF | 1332 | CX = MESSAGE BYTE COUNT |
| 662D E7E7FF | 1333 | MAXIMUM MESSAGE LENGTH IS 36 CHARACTERS |
| 662D E7E7FF | 1334 | ----- |
| 662D E7E7FF | 1335 | P_MSG PROC NEAR |
| 662D E7E7FF | 1336 | MOV AX:DATA |
| 662D E7E7FF | 1337 | MOV DS:AX |
| 662D E7E7FF | 1338 | CHP MSG_TST1.1 |
| 662D E7E7FF | 1339 | JNE G12 |
| 662D E7E7FF | 1340 | MOV DH:1 |
| 662D E7E7FF | 1341 | JMP ERP_BEOP |
| 662D E7E7FF | 1342 | G12: WRITE MSG: |
| 662D E7E7FF | 1343 | AL:CS:1511 |
| 662D E7E7FF | 1344 | INC SI |
| 662D E7E7FF | 1345 | MOV BH:0 |
| 662D E7E7FF | 1346 | MOV AH:14 |
| 662D E7E7FF | 1347 | INT 10H |
| 662D E7E7FF | 1348 | LOOP G12 |
| 662D E7E7FF | 1349 | MOV AX:0E00H |
| 662D E7E7FF | 1350 | INT 10H |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|-----------------|------|--|---------|------|--------|
| EA6C B80A0E | 1351 | MOV AX,BEAM | | 1426 | |
| EA6F C010 | 1352 | INT 10H | | 1429 | |
| EA71 C3 | 1353 | RET | | 1430 | |
| | 1354 | P_MIS ENDP | | 1431 | |
| | 1355 | 1000T STRAP LOADER | | 1432 | |
| | 1356 | 1000T STRAP LOADER | | 1433 | |
| | 1357 | IF A 5 1/4" DISKETTE DRIVE IS AVAILABLE | | 1434 | |
| | 1358 | ON THE SYSTEM, TRACK 0, SECTOR 1 IS READ INTO THE | | 1435 | |
| | 1359 | BOOT LOCATION (SECTOR 0, OFFSET 7C00) | | 1436 | |
| | 1360 | AND LOCATION IS TRANSFERRED THERE. | | 1437 | |
| | 1361 | | | 1438 | |
| | 1362 | IF THERE IS NO DISKETTE DRIVE, OR IF THERE IS | | 1439 | |
| | 1363 | IS A HARDWARE ERROR CONTROL IS TRANSFERRED | | 1440 | |
| | 1364 | TO THE CASSETTE BASIC ENTRY POINT. | | 1441 | |
| | 1365 | IPL ASSUMPTIONS | | 1442 | |
| | 1366 | 8255 PORT 45H BIT 0 | | 1443 | |
| | 1367 | = 1 IF IPL FROM DISKETTE | | 1444 | |
| | 1368 | ASSUME CS:CODE,DS:DATA | | 1445 | |
| | 1369 | BOOT_STRAP PROC NEAR | | 1446 | |
| EA72 | 1370 | BOOT_STRAP | | 1447 | |
| | 1371 | ENABLE INTERRUPTS | | 1448 | |
| | 1372 | ESTABLISH ADDRESSING | | 1449 | |
| EA75 FB | 1373 | MOV AX,DATA | | 1450 | |
| EA76 B80A00 | 1374 | MOV DS,AX | | 1451 | |
| EA78 800000 | 1375 | MOV AX,EQUIP_FLAG | | 1452 | |
| EA79 A10000 | 1376 | TEST AL,1 | | 1453 | |
| EA7B A001 | 1377 | JZ H3 | | 1454 | |
| EA7D 7A23 | 1378 | MUST LOAD SYSTEM FROM DISKETTE -- CX HAS RETRY COUNT | | 1455 | |
| | 1379 | | | 1456 | |
| | 1380 | | | 1457 | |
| EA7F B09400 | 1381 | MOV CX,4 | | 1458 | |
| E702 | 1382 | 1 SET RETRY COUNT | | 1459 | |
| E703 51 | 1383 | 1 IPL SYSTEM | | 1460 | |
| E705 8400 | 1384 | PUSH CX | | 1461 | |
| E706 C013 | 1385 | 1 SAVE RETRY COUNT | | 1462 | |
| E707 7214 | 1386 | 1 RESET THE DISKETTE SYSTEM | | 1463 | |
| E708 8402 | 1387 | DISKETTE_IO | | 1464 | |
| E709 800000 | 1388 | JC H2 | | 1465 | |
| E70E 8EC3 | 1389 | 1 IF ERROR, TRY AGAIN | | 1466 | |
| E710 B8007C | 1390 | MOV AH,2 | | 1467 | |
| E711 800100 | 1391 | MOV BX,0 | | 1468 | |
| E712 59 | 1392 | 1 DRIVE 0, HEAD 0 | | 1469 | |
| E713 7304 | 1393 | MOV CX,1 | | 1470 | |
| E714 E2E0 | 1394 | 1 SECTOR 1, TRACK 0 | | 1471 | |
| | 1395 | MOV AL,1 | | 1472 | |
| | 1396 | 1 READ ONE SECTOR | | 1473 | |
| | 1397 | INT 13H | | 1474 | |
| | 1398 | 1 DISKETTE_IO | | 1475 | |
| | 1399 | POP CX | | 1476 | |
| | 1400 | 1 RECOVER RETRY COUNT | | 1477 | |
| | 1401 | JNC H2 | | 1478 | |
| | 1402 | 1 CF SET BY UNSUCCESSFUL READ | | 1479 | |
| | 1403 | LOOP H1 | | 1480 | |
| | 1404 | 1 DO IT FOR RETRY TIMES | | 1481 | |
| | 1405 | | | 1482 | |
| E722 | 1406 | UNABLE TO IPL FROM THE DISKETTE | | 1483 | |
| E722 C01B | 1407 | | | 1484 | |
| | 1408 | | | 1485 | |
| | 1409 | | | 1486 | |
| E724 | 1410 | MOV JHP BOOT_LOCN | | 1487 | |
| E724 E4007C0000 | 1411 | ENDP | | 1488 | |
| | 1412 | 1000T STRAP LOADER | | 1489 | |
| | 1413 | 1000T STRAP LOADER | | 1490 | |
| | 1414 | 1000T STRAP LOADER | | 1491 | |
| | 1415 | 1000T STRAP LOADER | | 1492 | |
| | 1416 | 1000T STRAP LOADER | | 1493 | |
| | 1417 | 1000T STRAP LOADER | | 1494 | |
| | 1418 | 1000T STRAP LOADER | | 1495 | |
| | 1419 | 1000T STRAP LOADER | | 1496 | |
| | 1420 | 1000T STRAP LOADER | | 1497 | |
| | 1421 | 1000T STRAP LOADER | | 1498 | |
| | 1422 | 1000T STRAP LOADER | | 1499 | |
| | 1423 | 1000T STRAP LOADER | | 1500 | |
| | 1424 | 1000T STRAP LOADER | | 1501 | |
| | 1425 | 1000T STRAP LOADER | | 1502 | |
| | 1426 | 1000T STRAP LOADER | | 1503 | |
| | 1427 | 1000T STRAP LOADER | | 1504 | |

| LINE | LOC | OBJ | SOURCE |
|------|------|----------|--------|
| 1505 | | | |
| 1506 | 775F | | |
| 1507 | 775F | | |
| 1508 | 7761 | 7503 | |
| 1509 | 7763 | C8000 | |
| 1510 | 7766 | | |
| 1511 | 7766 | 59 | |
| 1512 | 7767 | 5F | |
| 1513 | 7768 | 5E | |
| 1514 | 7769 | 5A | |
| 1515 | 776A | 1F | |
| 1516 | 776B | CF | |
| 1517 | | | |
| 1518 | | | |
| 1519 | 776C | | |
| 1520 | | | |
| 1521 | 776E | 83C203 | |
| 1522 | 7771 | 8080 | |
| 1523 | 7773 | EE | |
| 1524 | | | |
| 1525 | | | |
| 1526 | | | |
| 1527 | 777A | 8A0A | |
| 1528 | 777B | 00C2 | |
| 1529 | 777B | 00C2 | |
| 1530 | 777A | 00C2 | |
| 1531 | 777C | 00C2 | |
| 1532 | 777E | 81E20E09 | |
| 1533 | 7782 | 8F78E7 | |
| 1534 | 7785 | 03FA | |
| 1535 | 7787 | 889A0000 | |
| 1536 | 778B | 42 | |
| 1537 | 7790 | EE | |
| 1538 | 7791 | 4A | |
| 1539 | 7792 | 2E8A05 | |
| 1540 | 7795 | EE | |
| 1541 | 7796 | 83C203 | |
| 1542 | 7799 | 8A0A | |
| 1543 | 779B | 241F | |
| 1544 | 779D | EE | |
| 1545 | 779E | 83E2A02 | |
| 1546 | 77A1 | 8080 | |
| 1547 | 77A3 | EE | |
| 1548 | 77A4 | E879 | |
| 1549 | | | |
| 1550 | | | |
| 1551 | | | |
| 1552 | | | |
| 1553 | 77A6 | 50 | |
| 1554 | 77A7 | 83C204 | |
| 1555 | 77A8 | 8003 | |
| 1556 | 77AC | EE | |
| 1557 | 77AD | 32C9 | |
| 1558 | 77AF | 83C202 | |
| 1559 | 77B2 | | |
| 1560 | 77B2 | EC | |
| 1561 | 77B3 | 8B20 | |
| 1562 | 77B5 | 7508 | |
| 1563 | 77B7 | E279 | |
| 1564 | 77B9 | 58 | |
| 1565 | 77BA | 80CCE0 | |
| 1566 | 77BD | E847 | |
| 1567 | 77BF | | |
| 1568 | 77BF | 08C9 | |
| 1569 | 77C1 | | |
| 1570 | 77C1 | EC | |
| 1571 | 77C2 | 4810 | |
| 1572 | 77C2 | 7508 | |
| 1573 | 77C9 | E279 | |
| 1574 | 77CB | 50 | |
| 1575 | 77CC | 80CCE0 | |
| 1576 | 77CC | E848 | |
| 1577 | 77CE | | |
| 1578 | 77CE | 4A | |
| 1579 | 77CF | 80C9 | |
| 1580 | 77CF | 80C9 | |
| 1581 | | | |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|---------|------|--------|---------|------|--------|
| E80E | 1736 | DB | E80B | 1736 | DB |
| E80F | 1737 | DB | E80C | 1737 | DB |
| E810 | 1738 | DB | E80D | 1738 | DB |
| E811 | 1739 | DB | E80E | 1739 | DB |
| E812 | 1740 | DB | E80F | 1740 | DB |
| E813 | 1741 | DB | E810 | 1741 | DB |
| E814 | 1742 | DB | E811 | 1742 | DB |
| E815 | 1743 | DB | E812 | 1743 | DB |
| E816 | 1744 | DB | E813 | 1744 | DB |
| E817 | 1745 | DB | E814 | 1745 | DB |
| E818 | 1746 | DB | E815 | 1746 | DB |
| E819 | 1747 | DB | E816 | 1747 | DB |
| E81A | 1748 | DB | E817 | 1748 | DB |
| E81B | 1749 | DB | E818 | 1749 | DB |
| E81C | 1750 | DB | E819 | 1750 | DB |
| E81D | 1751 | DB | E81A | 1751 | DB |
| E81E | 1752 | DB | E81B | 1752 | DB |
| E81F | 1753 | DB | E81C | 1753 | DB |
| E820 | 1754 | DB | E81D | 1754 | DB |
| E821 | 1755 | DB | E81E | 1755 | DB |
| E822 | 1756 | DB | E81F | 1756 | DB |
| E823 | 1757 | DB | E820 | 1757 | DB |
| E824 | 1758 | DB | E821 | 1758 | DB |
| E825 | 1759 | DB | E822 | 1759 | DB |
| E826 | 1760 | DB | E823 | 1760 | DB |
| E827 | 1761 | DB | E824 | 1761 | DB |
| E828 | 1762 | DB | E825 | 1762 | DB |
| E829 | 1763 | DB | E826 | 1763 | DB |
| E82A | 1764 | DB | E827 | 1764 | DB |
| E82B | 1765 | DB | E828 | 1765 | DB |
| E82C | 1766 | DB | E829 | 1766 | DB |
| E82D | 1767 | DB | E82A | 1767 | DB |
| E82E | 1768 | DB | E82B | 1768 | DB |
| E82F | 1769 | DB | E82C | 1769 | DB |
| E830 | 1770 | DB | E82D | 1770 | DB |
| E831 | 1771 | DB | E82E | 1771 | DB |
| E832 | 1772 | DB | E82F | 1772 | DB |
| E833 | 1773 | DB | E830 | 1773 | DB |
| E834 | 1774 | DB | E831 | 1774 | DB |
| E835 | 1775 | DB | E832 | 1775 | DB |
| E836 | 1776 | DB | E833 | 1776 | DB |
| E837 | 1777 | DB | E834 | 1777 | DB |
| E838 | 1778 | DB | E835 | 1778 | DB |
| E839 | 1779 | DB | E836 | 1779 | DB |
| E83A | 1780 | DB | E837 | 1780 | DB |
| E83B | 1781 | DB | E838 | 1781 | DB |
| E83C | 1782 | DB | E839 | 1782 | DB |
| E83D | 1783 | DB | E83A | 1783 | DB |
| E83E | 1784 | DB | E83B | 1784 | DB |
| E83F | 1785 | DB | E83C | 1785 | DB |
| E840 | 1786 | DB | E83D | 1786 | DB |
| E841 | 1787 | DB | E83E | 1787 | DB |
| E842 | 1788 | DB | E840 | 1788 | DB |
| E843 | 1789 | DB | E841 | 1789 | DB |
| E844 | 1790 | DB | E842 | 1790 | DB |
| E845 | 1791 | DB | E843 | 1791 | DB |
| E846 | 1792 | DB | E844 | 1792 | DB |
| E847 | 1793 | DB | E845 | 1793 | DB |
| E848 | 1794 | DB | E846 | 1794 | DB |
| E849 | 1795 | DB | E847 | 1795 | DB |
| E84A | 1796 | DB | E848 | 1796 | DB |
| E84B | 1797 | DB | E849 | 1797 | DB |
| E84C | 1798 | DB | E84A | 1798 | DB |
| E84D | 1799 | DB | E84B | 1799 | DB |
| E84E | 1800 | DB | E84C | 1800 | DB |
| E84F | 1801 | DB | E84D | 1801 | DB |
| E850 | 1802 | DB | E84E | 1802 | DB |
| E851 | 1803 | DB | E84F | 1803 | DB |
| E852 | 1804 | DB | E850 | 1804 | DB |
| E853 | 1805 | DB | E851 | 1805 | DB |
| E854 | 1806 | DB | E852 | 1806 | DB |
| E855 | 1807 | DB | E853 | 1807 | DB |
| E856 | 1808 | DB | E854 | 1808 | DB |
| E857 | 1809 | DB | E855 | 1809 | DB |
| E858 | 1810 | DB | E856 | 1810 | DB |
| E859 | 1811 | DB | E857 | 1811 | DB |
| E85A | 1812 | DB | E858 | 1812 | DB |
| E85B | 1813 | DB | E859 | 1813 | DB |
| E85C | 1814 | DB | E85A | 1814 | DB |
| E85D | 1815 | DB | E85B | 1815 | DB |
| E85E | 1816 | DB | E85C | 1816 | DB |
| E85F | 1817 | DB | E85D | 1817 | DB |
| E860 | 1818 | DB | E85E | 1818 | DB |
| E861 | 1819 | DB | E860 | 1819 | DB |
| E862 | 1820 | DB | E861 | 1820 | DB |
| E863 | 1821 | DB | E862 | 1821 | DB |
| E864 | 1822 | DB | E863 | 1822 | DB |
| E865 | 1823 | DB | E864 | 1823 | DB |
| E866 | 1824 | DB | E865 | 1824 | DB |
| E867 | 1825 | DB | E866 | 1825 | DB |
| E868 | 1826 | DB | E867 | 1826 | DB |
| E869 | 1827 | DB | E868 | 1827 | DB |
| E86A | 1828 | DB | E869 | 1828 | DB |
| E86B | 1829 | DB | E86A | 1829 | DB |
| E86C | 1830 | DB | E86B | 1830 | DB |
| E86D | 1831 | DB | E86C | 1831 | DB |
| E86E | 1832 | DB | E86D | 1832 | DB |
| E86F | 1833 | DB | E86E | 1833 | DB |
| E870 | 1834 | DB | E86F | 1834 | DB |
| E871 | 1835 | DB | E870 | 1835 | DB |
| E872 | 1836 | DB | E871 | 1836 | DB |
| E873 | 1837 | DB | E872 | 1837 | DB |
| E874 | 1838 | DB | E873 | 1838 | DB |
| E875 | 1839 | DB | E874 | 1839 | DB |
| E876 | 1840 | DB | E875 | 1840 | DB |
| E877 | 1841 | DB | E876 | 1841 | DB |
| E878 | 1842 | DB | E877 | 1842 | DB |
| E879 | 1843 | DB | E878 | 1843 | DB |
| E87A | 1844 | DB | E879 | 1844 | DB |
| E87B | 1845 | DB | E87A | 1845 | DB |
| E87C | 1846 | DB | E87B | 1846 | DB |
| E87D | 1847 | DB | E87C | 1847 | DB |
| E87E | 1848 | DB | E87D | 1848 | DB |
| E87F | 1849 | DB | E87E | 1849 | DB |
| E880 | 1850 | DB | E87F | 1850 | DB |
| E881 | 1851 | DB | E880 | 1851 | DB |
| E882 | 1852 | DB | E881 | 1852 | DB |
| E883 | 1853 | DB | E882 | 1853 | DB |
| E884 | 1854 | DB | E883 | 1854 | DB |
| E885 | 1855 | DB | E884 | 1855 | DB |
| E886 | 1856 | DB | E885 | 1856 | DB |
| E887 | 1857 | DB | E886 | 1857 | DB |
| E888 | 1858 | DB | E887 | 1858 | DB |
| E889 | 1859 | DB | E888 | 1859 | DB |
| E88A | 1860 | DB | E889 | 1860 | DB |
| E88B | 1861 | DB | E88A | 1861 | DB |
| E88C | 1862 | DB | E88B | 1862 | DB |
| E88D | 1863 | DB | E88C | 1863 | DB |
| E88E | 1864 | DB | E88D | 1864 | DB |
| E88F | 1865 | DB | E88E | 1865 | DB |
| E890 | 1866 | DB | E88F | 1866 | DB |
| E891 | 1867 | DB | E890 | 1867 | DB |
| E892 | 1868 | DB | E891 | 1868 | DB |
| E893 | 1869 | DB | E892 | 1869 | DB |
| E894 | 1870 | DB | E893 | 1870 | DB |
| E895 | 1871 | DB | E894 | 1871 | DB |
| E896 | 1872 | DB | E895 | 1872 | DB |
| E897 | 1873 | DB | E896 | 1873 | DB |
| E898 | 1874 | DB | E897 | 1874 | DB |
| E899 | 1875 | DB | E898 | 1875 | DB |
| E89A | 1876 | DB | E899 | 1876 | DB |
| E89B | 1877 | DB | E89A | 1877 | DB |
| E89C | 1878 | DB | E89B | 1878 | DB |
| E89D | 1879 | DB | E89C | 1879 | DB |
| E89E | 1880 | DB | E89D | 1880 | DB |
| E89F | 1881 | DB | E89E | 1881 | DB |
| E8A0 | 1882 | DB | E89F | 1882 | DB |
| E8A1 | 1883 | DB | E8A0 | 1883 | DB |
| E8A2 | 1884 | DB | E8A1 | 1884 | DB |
| E8A3 | 1885 | DB | E8A2 | 1885 | DB |
| E8A4 | 1886 | DB | E8A3 | 1886 | DB |
| E8A5 | 1887 | DB | E8A4 | 1887 | DB |
| E8A6 | 1888 | DB | E8A5 | 1888 | DB |
| E8A7 | 1889 | DB | E8A6 | 1889 | DB |
| E8A8 | 1890 | DB | E8A7 | 1890 | DB |
| E8A9 | 1891 | DB | E8A8 | 1891 | DB |
| E8AA | 1892 | DB | E8A9 | 1892 | DB |
| E8AB | 1893 | DB | E8AA | 1893 | DB |
| E8AC | 1894 | DB | E8AB | 1894 | DB |
| E8AD | 1895 | DB | E8AC | 1895 | DB |
| E8AE | 1896 | DB | E8AD | 1896 | DB |
| E8AF | 1897 | DB | E8AE | 1897 | DB |
| E8B0 | 1898 | DB | E8AF | 1898 | DB |
| E8B1 | 1899 | DB | E8B0 | 1899 | DB |
| E8B2 | 1900 | DB | E8B1 | 1900 | DB |
| E8B3 | 1901 | DB | E8B2 | 1901 | DB |
| E8B4 | 1902 | DB | E8B3 | 1902 | DB |
| E8B5 | 1903 | DB | E8B4 | 1903 | DB |
| E8B6 | 1904 | DB | E8B5 | 1904 | DB |
| E8B7 | 1905 | DB | E8B6 | 1905 | DB |
| E8B8 | 1906 | DB | E8B7 | 1906 | DB |
| E8B9 | 1907 | DB | E8B8 | 1907 | DB |
| E8BA | 1908 | DB | E8B9 | 1908 | DB |
| E8BB | 1909 | DB | E8BA | 1909 | DB |
| E8BC | 1910 | DB | E8BB | 1910 | DB |
| E8BD | 1911 | DB | E8BC | 1911 | DB |
| E8BE | 1912 | DB | E8BD | 1912 | DB |
| E8BF | 1913 | DB | E8BE | 1913 | DB |
| E8C0 | 1914 | DB | E8BF | 1914 | DB |
| E8C1 | 1915 | DB | E8C0 | 1915 | DB |
| E8C2 | 1916 | DB | E8C1 | 1916 | DB |
| E8C3 | 1917 | DB | E8C2 | 1917 | DB |
| E8C4 | 1918 | DB | E8C3 | 1918 | DB |
| E8C5 | 1919 | DB | E8C4 | 1919 | DB |
| E8C6 | 1920 | DB | E8C5 | 1920 | DB |
| E8C7 | 1921 | DB | E8C6 | 1921 | DB |
| E8C8 | 1922 | DB | E8C7 | 1922 | DB |
| E8C9 | 1923 | DB | E8C8 | 1923 | DB |
| E8CA | 1924 | DB | E8C9 | 1924 | DB |
| E8CB | 1925 | DB | E8CA | 1925 | DB |
| E8CC | 1926 | DB | E8CB | 1926 | DB |
| E8CD | 1927 | DB | E8CC | 1927 | DB |
| E8CE | 1928 | DB | E8CD | 1928 | DB |
| E8CF | 1929 | DB | E8CE | 1929 | DB |
| E8D0 | 1930 | DB | E8CF | 1930 | DB |
| E8D1 | 1931 | DB | E8D0 | 1931 | DB |
| E8D2 | 1932 | DB | E8D1 | 1932 | DB |
| E8D3 | 1933 | DB | E8D2 | 1933 | DB |
| E8D4 | 1934 | DB | E8D3 | 1934 | DB |
| E8D5 | 1935 | DB | E8D4 | 1935 | DB |
| E8D6 | 1936 | DB | E8D5 | 1936 | DB |
| E8D7 | 1937 | DB | E8D6 | 1937 | DB |
| E8D8 | 1938 | DB | E8D7 | 1938 | DB |
| E8D9 | 1939 | DB | E8D8 | 1939 | DB |
| E8DA | 1940 | DB | E8D9 | 1940 | DB |
| E8DB | 1941 | DB | E8DA | 1941 | DB |
| E8DC | 1942 | DB | E8DB | 1942 | DB |
| E8DD | 1943 | DB | E8DC | 1943 | DB |
| E8DE | 1944 | DB | E8DD | 1944 | DB |
| E8DF | 1945 | DB | E8DE | 1945 | DB |
| E8E0 | 1946 | DB | E8DF | 1946 | DB |
| E8E1 | 1947 | DB | E8E0 | 1947 | DB |
| E8E2 | 1948 | DB | E8E1 | 1948 | DB |
| E8E3 | 1949 | DB | E8E2 | 1949 | DB |
| E8E4 | 1950 | DB | E8E3 | 1950 | DB |
| E8E5 | 1951 | DB | E8E4 | 1951 | DB |
| E8E6 | 1952 | DB | E8E5 | 1952 | DB |
| E8E7 | 1953 | DB | E8E6 | 1953 | DB |
| E8E8 | 1954 | DB | E8E7 | 1954 | DB |
| E8E9 | 1955 | DB | E8E8 | 1955 | DB |
| E8EA | 1956 | DB | E8E9 | 1956 | DB |
| E8EB | 1957 | DB | E8EA | 1957 | DB |
| E8EC | 1958 | DB | E8EB | 1958 | DB |
| E8ED | 1959 | DB | E8EC | 1959 | DB |
| E8EE | 1960 | DB | E8ED | 1960 | DB |
| E8EF | 1961 | DB | E8EE | 1961 | DB |
| E8F0 | 1962 | DB | E8EF | 1962 | DB |
| E8F1 | 1963 | DB | E8F0 | 1963 | DB |
| E8F2 | 1964 | DB | E8F1 | 1964 | DB |
| E8F3 | 1965 | DB | E8F2 | 1965 | DB |
| E8F4 | 1966 | DB | E8F3 | 1966 | DB |
| E8F5 | 1967 | DB | E8F4 | 1967 | DB |
| E8F6 | 1968 | DB | E8F5 | 1968 | DB |
| E8F7 | 1969 | DB | E8F6 | 1969 | DB |
| E8F8 | 1970 | DB | E8F7 | 1970 | DB |
| E8F9 | 1971 | DB | E8F8 | 1971 | DB |
| E8FA | 1972 | DB | E8F9 | 1972 | DB |
| E8FB | 1973 | | | | |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|-----------|------|---|------------|------|--------------------|
| E994 800B | 1798 | MOV DS:AX | EAD0 80052 | 1875 | MOV AX,INS_KEY+256 |
| E994 8400 | 1799 | IN AL,XB_DATA | EAD0 80061 | 1876 | JMP K57 |
| E994 8400 | 1800 | PUSH AX | | | |
| E994 8401 | 1801 | IN AL,XB_CTL | | | |
| E994 8402 | 1802 | MOV AX,AL | | | |
| E994 8403 | 1803 | OR AL,AL | | | |
| E994 8404 | 1804 | OUT XB_CTL,AL | | | |
| E994 8405 | 1805 | XOR AL,AL | | | |
| E994 8406 | 1806 | OUT XB_CTL,AL | | | |
| E994 8407 | 1807 | MOV AX,AL | | | |
| E994 8408 | 1808 | POP AX | | | |
| E994 8409 | 1809 | TEST FOR OVERLAP SCAN CODE FROM KEYBOARD | | | |
| E994 8410 | 1810 | | | | |
| E994 8411 | 1811 | MOV AL,OFFH | | | |
| E994 8412 | 1812 | JNZ K16 | | | |
| E994 8413 | 1813 | JMP K62 | | | |
| E994 8414 | 1814 | | | | |
| E994 8415 | 1815 | TEST FOR SHIFT KEYS | | | |
| E994 8416 | 1816 | | | | |
| E994 8417 | 1817 | | | | |
| E994 8418 | 1818 | K16: AND AL,OFFH | | | |
| E994 8419 | 1819 | PUSH CS | | | |
| E994 8420 | 1820 | POP ES | | | |
| E994 8421 | 1821 | MOV DI,OFFSET K6 | | | |
| E994 8422 | 1822 | MOV CX,K6L | | | |
| E994 8423 | 1823 | REPNE SCASB | | | |
| E994 8424 | 1824 | | | | |
| E994 8425 | 1825 | MOV AL,AM | | | |
| E994 8426 | 1826 | JE K17 | | | |
| E994 8427 | 1827 | JMP K25 | | | |
| E994 8428 | 1828 | | | | |
| E994 8429 | 1829 | SHIFT KEY FOUND | | | |
| E994 8430 | 1830 | | | | |
| E994 8431 | 1831 | K17: SUB DI,OFFSET K6+1 | | | |
| E994 8432 | 1832 | MOV AH,CS-K71011 | | | |
| E994 8433 | 1833 | TEST AL,AM | | | |
| E994 8434 | 1834 | JNZ K23 | | | |
| E994 8435 | 1835 | | | | |
| E994 8436 | 1836 | SHIFT MAKE FOUND, DETERMINE SET OR TOGGLE | | | |
| E994 8437 | 1837 | | | | |
| E994 8438 | 1838 | MOV AX,SCROLL_SHIFT | | | |
| E994 8439 | 1839 | JAE K18 | | | |
| E994 8440 | 1840 | | | | |
| E994 8441 | 1841 | PLAIN SHIFT KEY, SET SHIFT ON | | | |
| E994 8442 | 1842 | | | | |
| E994 8443 | 1843 | OR KB_FLAG,AM | | | |
| E994 8444 | 1844 | JMP K26 | | | |
| E994 8445 | 1845 | | | | |
| E994 8446 | 1846 | TOGGLED SHIFT KEY, TEST FOR 1ST MAKE OR NOT | | | |
| E994 8447 | 1847 | | | | |
| E994 8448 | 1848 | K18: TEST KB_FLAG,CTL_SHIFT | | | |
| E994 8449 | 1849 | JNZ K25 | | | |
| E994 8450 | 1850 | MOV AX,INS_KEY | | | |
| E994 8451 | 1851 | TEST AL,INS_KEY | | | |
| E994 8452 | 1852 | JNZ K22 | | | |
| E994 8453 | 1853 | TEST KB_FLAG,ALT_SHIFT | | | |
| E994 8454 | 1854 | JZ K19 | | | |
| E994 8455 | 1855 | JMP K25 | | | |
| E994 8456 | 1856 | TEST KB_FLAG,NUM_STATE | | | |
| E994 8457 | 1857 | JNZ K21 | | | |
| E994 8458 | 1858 | TEST KB_FLAG,LEFT_SHIFT | | | |
| E994 8459 | 1859 | JZ K22 | | | |
| E994 8460 | 1860 | | | | |
| E994 8461 | 1861 | K20: MOV AX,520H | | | |
| E994 8462 | 1862 | JMP K57 | | | |
| E994 8463 | 1863 | | | | |
| E994 8464 | 1864 | K21: TEST KB_FLAG,LEFT_SHIFT+RIGHT_SHIFT | | | |
| E994 8465 | 1865 | JZ K20 | | | |
| E994 8466 | 1866 | | | | |
| E994 8467 | 1867 | K22: TEST AM,KB_FLAG,1 | | | |
| E994 8468 | 1868 | JNZ K26 | | | |
| E994 8469 | 1869 | OR KB_FLAG,1,AM | | | |
| E994 8470 | 1870 | XOR KB_FLAG,AM | | | |
| E994 8471 | 1871 | MOV AL,INS_KEY | | | |
| E994 8472 | 1872 | TEST AL,INS_KEY | | | |
| E994 8473 | 1873 | JNE K26 | | | |
| E994 8474 | 1874 | | | | |
| E994 8475 | 1875 | | | | |
| E994 8476 | 1876 | | | | |
| E994 8477 | 1877 | | | | |
| E994 8478 | 1878 | | | | |
| E994 8479 | 1879 | | | | |
| E994 8480 | 1880 | | | | |
| E994 8481 | 1881 | | | | |
| E994 8482 | 1882 | | | | |
| E994 8483 | 1883 | | | | |
| E994 8484 | 1884 | | | | |
| E994 8485 | 1885 | | | | |
| E994 8486 | 1886 | | | | |
| E994 8487 | 1887 | | | | |
| E994 8488 | 1888 | | | | |
| E994 8489 | 1889 | | | | |
| E994 8490 | 1890 | | | | |
| E994 8491 | 1891 | | | | |
| E994 8492 | 1892 | | | | |
| E994 8493 | 1893 | | | | |
| E994 8494 | 1894 | | | | |
| E994 8495 | 1895 | | | | |
| E994 8496 | 1896 | | | | |
| E994 8497 | 1897 | | | | |
| E994 8498 | 1898 | | | | |
| E994 8499 | 1899 | | | | |
| E994 8500 | 1900 | | | | |
| E994 8501 | 1901 | | | | |
| E994 8502 | 1902 | | | | |
| E994 8503 | 1903 | | | | |
| E994 8504 | 1904 | | | | |
| E994 8505 | 1905 | | | | |
| E994 8506 | 1906 | | | | |
| E994 8507 | 1907 | | | | |
| E994 8508 | 1908 | | | | |
| E994 8509 | 1909 | | | | |
| E994 8510 | 1910 | | | | |
| E994 8511 | 1911 | | | | |
| E994 8512 | 1912 | | | | |
| E994 8513 | 1913 | | | | |
| E994 8514 | 1914 | | | | |
| E994 8515 | 1915 | | | | |
| E994 8516 | 1916 | | | | |
| E994 8517 | 1917 | | | | |
| E994 8518 | 1918 | | | | |
| E994 8519 | 1919 | | | | |
| E994 8520 | 1920 | | | | |
| E994 8521 | 1921 | | | | |
| E994 8522 | 1922 | | | | |
| E994 8523 | 1923 | | | | |
| E994 8524 | 1924 | | | | |
| E994 8525 | 1925 | | | | |
| E994 8526 | 1926 | | | | |
| E994 8527 | 1927 | | | | |
| E994 8528 | 1928 | | | | |
| E994 8529 | 1929 | | | | |
| E994 8530 | 1930 | | | | |
| E994 8531 | 1931 | | | | |
| E994 8532 | 1932 | | | | |
| E994 8533 | 1933 | | | | |
| E994 8534 | 1934 | | | | |
| E994 8535 | 1935 | | | | |
| E994 8536 | 1936 | | | | |
| E994 8537 | 1937 | | | | |
| E994 8538 | 1938 | | | | |
| E994 8539 | 1939 | | | | |
| E994 8540 | 1940 | | | | |
| E994 8541 | 1941 | | | | |
| E994 8542 | 1942 | | | | |
| E994 8543 | 1943 | | | | |
| E994 8544 | 1944 | | | | |
| E994 8545 | 1945 | | | | |
| E994 8546 | 1946 | | | | |
| E994 8547 | 1947 | | | | |
| E994 8548 | 1948 | | | | |
| E994 8549 | 1949 | | | | |
| E994 8550 | 1950 | | | | |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|-----------------------|------|---|-----------------|--------|--|
| EAF1 474849 | 1951 | DN 71,72,73 | EB10 B01E00 | R 2326 | MOV BX,OFFSET KB_BUFFER ; RESET BUFFER TO EMPTY |
| EAF4 10111E1314151617 | 1952 | ----- SUPER-SHIFT-TABLE | EB13 491E1A00 | R 2327 | MOV BUFFER_HEAD,BX ; |
| EAF6 18191E1F20212223 | 1953 | DB 16,17,18,19,20,21,22,23 ; A-Z TYPEWRITER CHARS | EB17 491E1C00 | R 2328 | MOV BUFFER_TAIL,BX ; |
| EAF9 24252627282930 | 1954 | DB 24,25,26,27,28,29,30,31,32,33,34,35 | EB19 40A6710080 | R 2329 | MOV BIOS_BREAK,B0H ; TURN ON BIOS_BREAK BIT |
| EAFB 3637383940414243 | 1955 | DB 36,37,38,39,40,41,42,43,44,45,46,47,48 | EB20 C01B | R 2330 | INT 10H ; BREAK INTERRUPT VECTOR |
| EAFD 4950 | 1956 | DB 49,50 | EB22 800000 | R 2331 | MOV AX,0 ; PUT OUT DUMP CHARACTER |
| EAF1 474849 | 1957 | ----- IN ALTERNATE SHIFT, RESET NOT FOUND | EB25 F0A400 | R 2332 | JMP K57 ; BUFFER_FILL |
| EAF4 10111E1314151617 | 1958 | ----- ALTERNATE SHIFT, RESET NOT FOUND | EB29 | R 2333 | K39: ; NO-BREAK |
| EAF7 7512 | 1959 | K31: ; NO-RESET | EB2B 3C45 | R 2334 | CHP AL,MPM_KEY ; LOOK FOR PAUSE KEY |
| EAF9 7505 | 1960 | CHP AL,57 ; TEST FOR SPACE KEY | EB2E 7521 | R 2335 | JNE K41 ; NO-PAUSE |
| EAFB 7505 | 1961 | JNE K32 ; NOT THERE | EB30 000E100000 | R 2336 | DB KB_FLAG_1,HOLD_STATE ; TURN ON THE HOLD FLAG |
| EAFD 8020 | 1962 | MOV AL,' ' ; SET SPACE CHAR | EB31 8020 | R 2337 | MOV AL,EOI ; END OF INTERRUPT TO CONTROL PORT |
| EAF1 474849 | 1963 | JMP K57 ; BUFFER_FILL | EB33 6A20 | R 2338 | OUT 020H,AL ; ALLOW FURTHER KEYSTROKE INTS |
| EAF4 10111E1314151617 | 1964 | ----- LOOK FOR KEY PAD ENTRY | EB35 | R 2339 | ----- DURING PAUSE INTERVAL, TURN CRT BACK ON |
| EAF7 7512 | 1965 | K32: ; ALT-KEY-PAD | EB37 403E490007 | R 2340 | CHP CRY,MODE.7 ; IS THIS BLACK AND WHITE CARD |
| EAF9 7505 | 1966 | MOV DI,OFFSET K30 | EB3A 7407 | R 2341 | JE K40 ; YES, NOTHING TO DO |
| EAFB 7505 | 1967 | MOV CX,10 ; LOOK FOR ENTRY USING KEYPAD | EB3C 0A0503 | R 2342 | MOV DX,0300H ; PORT FOR COLOR CARD |
| EAFD 7505 | 1968 | REPNE SCASB ; LOOK FOR MATCH | EB3F 0A0500 | R 2343 | MOV AL,CPT_MODE_SET ; GET THE VALUE OF THE CURRENT MODE |
| EAF1 474849 | 1969 | JNE K33 ; NO-ALT-KEYPAD | EB42 EE | R 2344 | OUT DX,AL ; SET THE CRT MODE, SO THAT CRT IS ON |
| EAF4 10111E1314151617 | 1970 | SUB DI,OFFSET K30+1 | EB43 | R 2345 | TEST KB_FLAG_1,HOLD_STATE ; PAUSE-LOOP |
| EAF6 18191E1F20212223 | 1971 | MOV AL,ALT_INPUT | EB45 75F9 | R 2346 | JNZ K40 ; LOOP UNTIL FLAG TURNED OFF |
| EAF8 24252627282930 | 1972 | MOV AH,10 ; MULTIPLY BY 10 | EB48 E70A0F | R 2347 | JMP K27 ; INTERRUPT_RETURN,NO_EOI |
| EAFB 3637383940414243 | 1973 | MOV AH,10 ; MULTIPLY BY 10 | EB4B | R 2348 | K41: ; NO-PAUSE |
| EAFD 4950 | 1974 | MOV AH,10 ; MULTIPLY BY 10 | EB4D | R 2349 | ----- TEST SPECIAL CASE KEY 55 |
| EAF1 474849 | 1975 | MOV AH,10 ; MULTIPLY BY 10 | EB4F 7506 | R 2350 | CHP AL,55 ; NOT-KEY-55 |
| EAF4 10111E1314151617 | 1976 | MOV AH,10 ; MULTIPLY BY 10 | EB51 000072 | R 2351 | JNE K42 ; NOT-KEY-55 |
| EAF6 18191E1F20212223 | 1977 | MOV AH,10 ; MULTIPLY BY 10 | EB53 F08500 | R 2352 | MOV AX,114+256 ; BUFFER_FILL |
| EAF8 24252627282930 | 1978 | MOV AH,10 ; MULTIPLY BY 10 | EB55 | R 2353 | JMP K57 ; |
| EAFB 3637383940414243 | 1979 | MOV AH,10 ; MULTIPLY BY 10 | EB57 | R 2354 | ----- SET UP TO TRANSLATE CONTROL SHIFT |
| EAFD 4950 | 1980 | MOV AH,10 ; MULTIPLY BY 10 | EB59 | R 2355 | K42: ; NOT-KEY-55 |
| EAF1 474849 | 1981 | MOV AH,10 ; MULTIPLY BY 10 | EB5B 8892E8 | R 2356 | MOV BX,OFFSET KB ; SET UP TO TRANSLATE CTL |
| EAF4 10111E1314151617 | 1982 | MOV AH,10 ; MULTIPLY BY 10 | EB5E 3C3B | R 2357 | CHP AL,59 ; IS IT IN TABLE |
| EAF6 18191E1F20212223 | 1983 | MOV AH,10 ; MULTIPLY BY 10 | EB60 7303 | R 2358 | JAE K43 ; CTL-TABLE-TRANSLATE |
| EAF8 24252627282930 | 1984 | MOV AH,10 ; MULTIPLY BY 10 | EB62 E7890 | R 2359 | JMP K56 ; YES, DO TRANSLATE CHAR |
| EAFB 3637383940414243 | 1985 | MOV AH,10 ; MULTIPLY BY 10 | EB64 | R 2360 | K43: ; CTL-TABLE-TRANSLATE |
| EAFD 4950 | 1986 | MOV AH,10 ; MULTIPLY BY 10 | EB66 B0CCEB | R 2361 | MOV BX,OFFSET K9 ; CTL TABLE SCAN |
| EAF1 474849 | 1987 | MOV AH,10 ; MULTIPLY BY 10 | EB68 E7C300 | R 2362 | JMP K43 ; TRANSLATE_SCAN |
| EAF4 10111E1314151617 | 1988 | MOV AH,10 ; MULTIPLY BY 10 | EB6A | R 2363 | ----- NOT IN CONTROL SHIFT |
| EAF6 18191E1F20212223 | 1989 | MOV AH,10 ; MULTIPLY BY 10 | EB6C | R 2364 | K44: ; NOT-CTL-SHIFT |
| EAF8 24252627282930 | 1990 | MOV AH,10 ; MULTIPLY BY 10 | EB6E 7506 | R 2365 | CHP AL,71 ; TEST FOR KEYPAD REGION |
| EAFB 3637383940414243 | 1991 | MOV AH,10 ; MULTIPLY BY 10 | EB6F 7320 | R 2366 | JAE K48 ; HANDLE KEYPAD REGION |
| EAFD 4950 | 1992 | MOV AH,10 ; MULTIPLY BY 10 | EB71 7450 | R 2367 | TEST KB_FLAG_LEFT_SHIFT+RIGHT_SHIFT ; TEST FOR SHIFT STATE |
| EAF1 474849 | 1993 | MOV AH,10 ; MULTIPLY BY 10 | EB73 | R 2368 | JZ K54 ; |
| EAF4 10111E1314151617 | 1994 | MOV AH,10 ; MULTIPLY BY 10 | EB75 | R 2369 | ----- UPPER CASE, HANDLE SPECIAL CASES |
| EAF6 18191E1F20212223 | 1995 | MOV AH,10 ; MULTIPLY BY 10 | EB77 3C0F | R 2370 | CHP AL,15 ; BACK TAB KEY |
| EAF8 24252627282930 | 1996 | MOV AH,10 ; MULTIPLY BY 10 | EB79 7506 | R 2371 | CHP K45 ; NOT-BACK-TAB |
| EAFB 3637383940414243 | 1997 | MOV AH,10 ; MULTIPLY BY 10 | EB7B 80000F | R 2372 | MOV AX,15+256 ; SET PSEUDO SCAN CODE |
| EAFD 4950 | 1998 | MOV AH,10 ; MULTIPLY BY 10 | EB7D E86190 | R 2373 | JMP K57 ; BUFFER_FILL |
| EAF1 474849 | 1999 | MOV AH,10 ; MULTIPLY BY 10 | EB7F | R 2374 | K45: ; NOT-BACK-TAB |
| EAF4 10111E1314151617 | 2000 | MOV AH,10 ; MULTIPLY BY 10 | EB81 3C37 | R 2375 | CHP AL,55 ; PRINT SCREEN KEY |
| EAF6 18191E1F20212223 | 2001 | MOV AH,10 ; MULTIPLY BY 10 | EB83 7509 | R 2376 | JNE K46 ; NOT-PRINT-SCREEN |
| EAF8 24252627282930 | 2002 | MOV AH,10 ; MULTIPLY BY 10 | EB85 | R 2377 | ----- ISSUE INTERRUPT TO INDICATE PRINT SCREEN FUNCTION |
| EAFB 3637383940414243 | 2003 | MOV AH,10 ; MULTIPLY BY 10 | EB87 | R 2378 | K46: ; NOT-PRINT-SCREEN |
| EAFD 4950 | 2004 | MOV AH,10 ; MULTIPLY BY 10 | EB89 | R 2379 | MOV AL,EOI ; END OF CURRENT INTERRUPT |
| EAF1 474849 | 2005 | MOV AH,10 ; MULTIPLY BY 10 | EB8B 8020 | R 2380 | OUT 020H,AL ; NO FURTHER THINGS CAN HAPPEN |
| EAF4 10111E1314151617 | 2006 | MOV AH,10 ; MULTIPLY BY 10 | EB8D C005 | R 2381 | INT 5H ; ISSUE PRINT SCREEN INTERRUPT |
| EAF6 18191E1F20212223 | 2007 | MOV AH,10 ; MULTIPLY BY 10 | EB8F E040FE | R 2382 | JMP K27 ; GO BACK WITHOUT EOI OCCURRING |
| EAF8 24252627282930 | 2008 | MOV AH,10 ; MULTIPLY BY 10 | EB91 | R 2383 | K46: ; NOT-PRINT-SCREEN |
| EAFB 3637383940414243 | 2009 | MOV AH,10 ; MULTIPLY BY 10 | EB93 | R 2384 | MOV AL,59 ; FUNCTION KEYS |
| EAFD 4950 | 2010 | MOV AH,10 ; MULTIPLY BY 10 | EB95 3C3B | R 2385 | CHP AL,59 ; NOT-UPPER-FUNCTION |
| EAF1 474849 | 2011 | MOV AH,10 ; MULTIPLY BY 10 | EB97 7506 | R 2386 | MOV AX,114+256 ; UPPER CASE PSEUDO SCAN CODES |
| EAF4 10111E1314151617 | 2012 | MOV AH,10 ; MULTIPLY BY 10 | EB99 | R 2387 | JMP K47 ; |
| EAF6 18191E1F20212223 | 2013 | MOV AH,10 ; MULTIPLY BY 10 | EB9B 7506 | R 2388 | MOV AX,114+256 ; UPPER CASE PSEUDO SCAN CODES |
| EAF8 24252627282930 | 2014 | MOV AH,10 ; MULTIPLY BY 10 | EB9D | R 2389 | JMP K47 ; |
| EAFB 3637383940414243 | 2015 | MOV AH,10 ; MULTIPLY BY 10 | EB9F | R 2390 | ----- |
| EAFD 4950 | 2016 | MOV AH,10 ; MULTIPLY BY 10 | EBA1 | R 2391 | K30: ; NOT-ALT-SHIFT |
| EAF1 474849 | 2017 | MOV AH,10 ; MULTIPLY BY 10 | EBA3 | R 2392 | TEST KB_FLAG_CTL_SHIFT ; ARE WE IN CONTROL SHIFT |
| EAF4 10111E1314151617 | 2018 | MOV AH,10 ; MULTIPLY BY 10 | EBA5 | R 2393 | JZ K44 ; NOT-CTL-SHIFT |
| EAF6 18191E1F20212223 | 2019 | MOV AH,10 ; MULTIPLY BY 10 | EBA7 | R 2394 | ----- CONTROL SHIFT, TEST SPECIAL CHARACTERS |
| EAF8 24252627282930 | 2020 | MOV AH,10 ; MULTIPLY BY 10 | EBA9 | R 2395 | K31: ; NOT-ALT-SHIFT |
| EAFB 3637383940414243 | 2021 | MOV AH,10 ; MULTIPLY BY 10 | EBAB | R 2396 | TEST KB_FLAG_CTL_SHIFT ; ARE WE IN CONTROL SHIFT |
| EAFD 4950 | 2022 | MOV AH,10 ; MULTIPLY BY 10 | EBAD | R 2397 | JZ K44 ; NOT-CTL-SHIFT |
| EAF1 474849 | 2023 | MOV AH,10 ; MULTIPLY BY 10 | EBAF | R 2398 | ----- TEST FOR BREAK AND PAUSE KEYS |
| EAF4 10111E1314151617 | 2024 | MOV AH,10 ; MULTIPLY BY 10 | EBB1 | R 2399 | K32: ; NOT-ALT-SHIFT |
| EAF6 18191E1F20212223 | 2025 | MOV AH,10 ; MULTIPLY BY 10 | EBB3 | R 2400 | TEST KB_FLAG_CTL_SHIFT ; ARE WE IN CONTROL SHIFT |
| EAF8 24252627282930 | 2026 | MOV AH,10 ; MULTIPLY BY 10 | EBB5 | R 2401 | JZ K44 ; NOT-CTL-SHIFT |
| EAFB 3637383940414243 | 2027 | MOV AH,10 ; MULTIPLY BY 10 | EBB7 | R 2402 | ----- |
| EAFD 4950 | 2028 | MOV AH,10 ; MULTIPLY BY 10 | EBB9 | R 2403 | K33: ; NOT-ALT-SHIFT |
| EAF1 474849 | 2029 | MOV AH,10 ; MULTIPLY BY 10 | EBBB | R 2404 | TEST KB_FLAG_CTL_SHIFT ; ARE WE IN CONTROL SHIFT |
| EAF4 10111E1314151617 | 2030 | MOV AH,10 ; MULTIPLY BY 10 | EBBD | R 2405 | JZ K44 ; NOT-CTL-SHIFT |
| EAF6 18191E1F20212223 | 2031 | MOV AH,10 ; MULTIPLY BY 10 | EBBF | R 2406 | ----- |
| EAF8 24252627282930 | 2032 | MOV AH,10 ; MULTIPLY BY 10 | EBC1 | R 2407 | K34: ; NOT-ALT-SHIFT |
| EAFB 3637383940414243 | 2033 | MOV AH,10 ; MULTIPLY BY 10 | EBC3 | R 2408 | TEST KB_FLAG_CTL_SHIFT ; ARE WE IN CONTROL SHIFT |
| EAFD 4950 | 2034 | MOV AH,10 ; MULTIPLY BY 10 | EBC5 | R 2409 | JZ K44 ; NOT-CTL-SHIFT |
| EAF1 474849 | 2035 | MOV AH,10 ; MULTIPLY BY 10 | EBC7 | R 2410 | ----- |
| EAF4 10111E1314151617 | 2036 | MOV AH,10 ; MULTIPLY BY 10 | EBC9 | R 2411 | K35: ; NOT-ALT-SHIFT |
| EAF6 18191E1F20212223 | 2037 | MOV AH,10 ; MULTIPLY BY 10 | EBCB | R 2412 | TEST KB_FLAG_CTL_SHIFT ; ARE WE IN CONTROL SHIFT |
| EAF8 24252627282930 | 2038 | MOV AH,10 ; MULTIPLY BY 10 | EBCD | R 2413 | JZ K44 ; NOT-CTL-SHIFT |
| EAFB 3637383940414243 | 2039 | MOV AH,10 ; MULTIPLY BY 10 | EBCE | R 2414 | ----- |
| EAFD 4950 | 2040 | MOV AH,10 ; MULTIPLY BY 10 | EBD0 | R 2415 | K36: ; NOT-ALT-SHIFT |
| EAF1 474849 | 2041 | MOV AH,10 ; MULTIPLY BY 10 | EBD2 | R 2416 | TEST KB_FLAG_CTL_SHIFT ; ARE WE IN CONTROL SHIFT |
| EAF4 10111E1314151617 | 2042 | MOV AH,10 ; MULTIPLY BY 10 | EBD4 | R 2417 | JZ K44 ; NOT-CTL-SHIFT |
| EAF6 18191E1F20212223 | 2043 | MOV AH,10 ; MULTIPLY BY 10 | EBD6 | R 2418 | ----- |
| EAF8 24252627282930 | 2044 | MOV AH,10 ; MULTIPLY BY 10 | EBD8 | R 2419 | K37: ; NOT-ALT-SHIFT |
| EAFB 3637383940414243 | 2045 | MOV AH,10 ; MULTIPLY BY 10 | EBDA | R 2420 | TEST KB_FLAG_CTL_SHIFT ; ARE WE IN CONTROL SHIFT |
| EAFD 4950 | 2046 | MOV AH,10 ; MULTIPLY BY 10 | EBDC | R 2421 | JZ K44 ; NOT-CTL-SHIFT |
| EAF1 474849 | 2047 | MOV AH,10 ; MULTIPLY BY 10 | EBDE | R 2422 | ----- |
| EAF4 10111E1314151617 | 2048 | MOV AH,10 ; MULTIPLY BY 10 | EBE0 | R 2423 | K38: ; NOT-ALT-SHIFT |
| EAF6 18191E1F20212223 | 2049 | MOV AH,10 ; MULTIPLY BY 10 | EBE2 | R 2424 | TEST KB_FLAG_CTL_SHIFT ; ARE WE IN CONTROL SHIFT |
| EAF8 24252627282930 | 2050 | MOV AH,10 ; MULTIPLY BY 10 | EBE4 | R 2425 | JZ K44 ; NOT-CTL-SHIFT |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|-----------------|--------|--------------------------------------|-----------|------|--------|
| EB90 E99700 | 2103 | JMP K63 | EBF1 740F | 2180 | JZ K68 |
| EB93 | 2104 | | | 2181 | ; |
| EB93 801FE9 | 2105 | K67: MOV BX,OFFSET K11 | | 2182 | ; |
| EB96 EB40 | 2106 | JMP SHORT K56 | | 2183 | ; |
| | 2107 | | | 2184 | ; |
| | 2108 | | | 2185 | ; |
| | 2109 | | | 2186 | ; |
| | 2110 | | | 2187 | ; |
| EB96 | 2111 | | | 2188 | ; |
| EB96 F606170020 | R 2112 | TEST KB_FLAG-NMPL_STATE | | 2189 | ; |
| EB9D 7520 | 2113 | JNZ K52 | | 2190 | ; |
| EB9F F606170003 | R 2114 | TEST KB_FLAG-LEFT_SHIFT-ORIENT_SHIFT | | 2191 | ; |
| EBAA 7520 | 2115 | JNZ K53 | | 2192 | ; |
| | 2116 | | | 2193 | ; |
| | 2117 | | | 2194 | ; |
| | 2118 | | | 2195 | ; |
| EBAA | 2119 | | | 2196 | ; |
| EBAA 3C4A | 2120 | CHP AL,74 | | 2197 | ; |
| EBAA 740D | 2121 | JE K50 | | 2198 | ; |
| EBAA 3C4E | 2122 | CHP AL,7B | | 2199 | ; |
| EBAC 740C | 2123 | JE K51 | | 2200 | ; |
| EBAC 3C47 | 2124 | SUB AL,71 | | 2201 | ; |
| EB00 B07AE9 | 2125 | MOV BX,OFFSET K15 | | 2202 | ; |
| EB03 EB77 | 2126 | JMP SHORT K64 | | 2203 | ; |
| | 2127 | | | 2204 | ; |
| EB05 B0204A | 2128 | MOV AX,74+256+... | | 2205 | ; |
| EB06 EB2E | 2129 | JMP SHORT K57 | | 2206 | ; |
| | 2130 | | | 2207 | ; |
| EB0A B0204E | 2131 | MOV AX,70+256+... | | 2208 | ; |
| EB0A EB1D | 2132 | JMP SHORT K57 | | 2209 | ; |
| | 2133 | | | 2210 | ; |
| | 2134 | | | 2211 | ; |
| | 2135 | | | 2212 | ; |
| | 2136 | | | 2213 | ; |
| EB0F | 2137 | | | 2214 | ; |
| EB0F F606170003 | R 2138 | TEST KB_FLAG-LEFT_SHIFT-ORIENT_SHIFT | | 2215 | ; |
| EB0C 7520 | 2139 | JNZ K49 | | 2216 | ; |
| | 2140 | | | 2217 | ; |
| EB0C | 2141 | | | 2218 | ; |
| EB0C 3C46 | 2142 | SUB AL,70 | | 2219 | ; |
| EB0C B0A0E9 | 2143 | MOV BX,OFFSET K14 | | 2220 | ; |
| EB0C EB0B | 2144 | JMP SHORT K56 | | 2221 | ; |
| | 2145 | | | 2222 | ; |
| | 2146 | | | 2223 | ; |
| EB0C | 2147 | | | 2224 | ; |
| EB0C 3C3B | 2148 | CHP AL,59 | | 2225 | ; |
| EB0F 720A | 2149 | JB K55 | | 2226 | ; |
| EB01 B00D | 2150 | MOV AL,0 | | 2227 | ; |
| EB03 EB07 | 2151 | JMP SHORT K57 | | 2228 | ; |
| | 2152 | | | 2229 | ; |
| EB05 | 2153 | | | 2230 | ; |
| EB05 B0E2E6 | R 2154 | MOV BX,OFFSET K10 | | 2231 | ; |
| | 2155 | | | 2232 | ; |
| | 2156 | | | 2233 | ; |
| EB0B | 2157 | | | 2234 | ; |
| EB0B FFCB | 2158 | | | 2235 | ; |
| EB0A 2E07 | 2159 | DEC AL | | 2236 | ; |
| | 2160 | | | 2237 | ; |
| | 2161 | | | 2238 | ; |
| | 2162 | | | 2239 | ; |
| | 2163 | | | 2240 | ; |
| | 2164 | | | 2241 | ; |
| EB0C | 2165 | | | 2242 | ; |
| EB0C 3C7F | 2166 | CHP AL,-1 | | 2243 | ; |
| EB0E 741F | 2167 | JE K59 | | 2244 | ; |
| EB0C B0FCFF | 2168 | CHP AN,-1 | | 2245 | ; |
| EB03 741A | 2169 | JE K59 | | 2246 | ; |
| | 2170 | | | 2247 | ; |
| | 2171 | | | 2248 | ; |
| EB05 | 2172 | | | 2249 | ; |
| EB05 F606170040 | R 2173 | TEST KB_FLAG-CAPS_STATE | | 2250 | ; |
| EB0A 7420 | 2174 | JZ K61 | | 2251 | ; |
| | 2175 | | | 2252 | ; |
| | 2176 | | | 2253 | ; |
| | 2177 | | | 2254 | ; |
| EB0C F606170003 | R 2178 | TEST KB_FLAG-LEFT_SHIFT-ORIENT_SHIFT | | 2255 | ; |
| | 2179 | | | 2256 | ; |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|---------|------|------------------|---------|------|--|
| | 2255 | 1-- INT 13 ----- | | 2332 | MOV DH,AL ; SAVE # SECTORS IN DH |
| | 2256 | DISKETTE 1/0 | | 2333 | AND AH,0 ; INDICATE A READ OPERATION |
| | 2257 | INPUT | | 2334 | OR AH,AH ; AH=0 |
| | 2258 | | | 2335 | JZ DISK_RESET ; AH=1 |
| | 2259 | | | 2336 | DEC AH |
| | 2260 | | | 2337 | JZ DISK_STATUS |
| | 2261 | | | 2338 | MOV DISKETTE_STATUS,0 ; RESET THE STATUS INDICATOR |
| | 2262 | | | 2339 | CHP DL,4 ; TEST FOR DRIVE IN 0-3 RANGE |
| | 2263 | | | 2340 | JAE J3 ; ERROR IF ABOVE |
| | 2264 | | | 2341 | DEC AH |
| | 2265 | | | 2342 | JZ DISK_READ |
| | 2266 | | | 2343 | DEC AH |
| | 2267 | | | 2344 | JNZ ; TEST_DISK_VRF |
| | 2268 | | | | |
| | 2269 | | | | |
| | 2270 | | | 2345 | JMP DISK_WRITE |
| | 2271 | | | 2346 | JZ ; TEST_DISK_VRF |
| | 2272 | | | | |
| | 2273 | | | 2347 | DEC AH ; AH=4 |
| | 2274 | | | 2348 | JZ DISK_VRF |
| | 2275 | | | 2349 | DEC AH ; AH=5 |
| | 2276 | | | 2350 | JZ DISK_FORMAT |
| | 2277 | | | 2351 | J3 ; BAD_COMMAND |
| | 2278 | | | 2352 | MOV DISKETTE_STATUS,BAD_CMD ; ERROR CODE, NO SECTORS TRANSFERRED |
| | 2279 | | | 2353 | RET ; UNDEFINED OPERATION |
| | 2280 | | | 2354 | J1 ENDP |
| | 2281 | | | 2355 | |
| | 2282 | | | 2356 | !----- RESET THE DISKETTE SYSTEM |
| | 2283 | | | 2357 | |
| | 2284 | | | | |
| | 2285 | | | 2358 | DISK_RESET PROC NEAR |
| | 2286 | | | 2359 | MOV DX,03F0H ; ADAPTER CONTROL PORT |
| | 2287 | | | 2360 | CLI ; NO INTERRUPTS |
| | 2288 | | | 2361 | MOV AL,MOTOR_STATUS ; MOTOR IS ON |
| | 2289 | | | 2362 | MOV CL,4 ; SHIFT COUNT |
| | 2290 | | | 2363 | SAL AL,CL ; MOVE MOTOR VALUE TO HIGH NIBBLE |
| | 2291 | | | 2364 | TEST AL,20H ; SELECT CORRESPONDING DRIVE |
| | 2292 | | | 2365 | JNZ J5 ; JUMP IF MOTOR ONE IS ON |
| | 2293 | | | 2366 | TEST AL,40H ; JUMP IF MOTOR TWO IS ON |
| | 2294 | | | 2367 | JNZ J6 ; JUMP IF MOTOR ZERO IS ON |
| | 2295 | | | 2368 | TEST AL,80H ; JUMP IF MOTOR ZERO IS ON |
| | 2296 | | | 2369 | JC J6 |
| | 2297 | | | 2370 | JNC AL |
| | 2298 | | | 2371 | JNC AL |
| | 2299 | | | 2372 | JNC AL |
| | 2300 | | | 2373 | JNC AL |
| | 2301 | | | 2374 | OUT DX,AL ; TURN ON INTERRUPT ENABLE |
| | 2302 | | | 2375 | MOV SEEK_STATUS,0 ; RESET THE ADAPTER |
| | 2303 | | | 2376 | MOV DISKETTE_STATUS,0 ; SET RECAL REQUIRED ON ALL DRIVES |
| | 2304 | | | 2377 | OR AL,4 ; SET OK STATUS FOR DISKETTE |
| | 2305 | | | 2378 | OUT DX,AL ; TURN OFF THE RESET |
| | 2306 | | | 2379 | STI ; REENABLE THE INTERRUPTS |
| | 2307 | | | 2380 | CALL CHK_STAT_2 ; DO SENSE INTERRUPT STATUS FOLLOWING RESET |
| | 2308 | | | 2381 | MOV AL,NEC_STATUS ; IGNORE ERROR RETURN AND DO DMA TEST |
| | 2309 | | | 2382 | CHP AL,00H ; TEST FOR DRIVE READY TRANSITION |
| | 2310 | | | 2383 | JZ J7 ; EVERYTHING OK |
| | 2311 | | | 2384 | OR DISKETTE_STATUS,BAD_NEC ; SET ERROR CODE |
| | 2312 | | | 2385 | SHORT J8 ; RESET_RET |
| | 2313 | | | 2386 | |
| | 2314 | | | 2387 | !----- SEND SPECIFY COMMAND TO NEC |
| | 2315 | | | 2388 | |
| | 2316 | | | 2389 | J7 ; DRIVE_READY |
| | 2317 | | | 2390 | MOV AH,03H ; SPECIFY COMMAND |
| | 2318 | | | 2391 | CALL NEC_OUTPUT ; OUTPUT THE COMMAND |
| | 2319 | | | 2392 | CALL BX,1 ; FIRST BYTE PARAM IN BLOCK |
| | 2320 | | | 2393 | CALL GET_PARH ; TO THE NEC CONTROLLER |
| | 2321 | | | 2394 | MOV BX,3 ; SECOND BYTE PARAM IN BLOCK |
| | 2322 | | | 2395 | CALL GET_PARH ; TO THE NEC CONTROLLER |
| | 2323 | | | 2396 | J8 ; RETURN TO CALLER |
| | 2324 | | | 2397 | RET |
| | 2325 | | | 2398 | DISK_RESET ENDP |
| | 2326 | | | 2399 | |
| | 2327 | | | 2400 | !----- DISKETTE STATUS ROUTINE |
| | 2328 | | | 2401 | |
| | 2329 | | | 2402 | DISK_STATUS PROC NEAR |
| | 2330 | | | 2403 | MOV AL,DISKETTE_STATUS |
| | 2331 | | | 2404 | RET |
| | 2332 | | | 2405 | DISK_STATUS ENDP |
| | 2333 | | | 2406 | |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|---------|--------|---|---------|------|--|
| | 2497 | DISKETTE READ | | 2484 | POP DX ; RECOVER REGISTERS |
| | 2498 | DISK_READ PROC NEAR | | 2485 | ----- WAIT FOR MOTOR IF WRITE OPERATION |
| | 2499 | MOV AL,040H | | 2486 | TEST MOTOR_STATUS,B0H ; IS THIS A WRITE |
| | 2500 | J9: CALL DPA_SETUP | | 2487 | JZ J16 ; NO, CONTINUE WITHOUT WAIT |
| | 2501 | MOV AH,040H | | 2488 | MOV BX,20 ; GET THE MOTOR WAIT |
| | 2502 | DISK_READ PROC NEAR | | 2489 | CALL GET_PARM ; PARAMETER |
| | 2503 | MOV AX,SHORT RH_OPEN | | 2490 | OR AX,AX ; TEST FOR NO WAIT |
| | 2504 | ENDP | | 2491 | J12: JZ J16 ; TEST WAIT TIME |
| | 2505 | DISKETTE VERIFY | | 2492 | MOV CX,0 ; SET UP 1/8 SECOND LOOP TIME |
| | 2506 | DISK_VERIFY PROC NEAR | | 2493 | LOOP J13 ; WAIT FOR THE REQUIRED TIME |
| | 2507 | MOV AL,040H | | 2494 | DEC AH ; DECREMENT THE VALUE |
| | 2508 | J9: JZ J16 | | 2495 | JMP J1C ; ARE WE DONE YET |
| | 2509 | DISK_VERIFY ENDP | | 2496 | ----- MOTOR RUNNING |
| | 2510 | DISKETTE FORMAT | | 2497 | STI ; INTERRUPTS BACK ON FOR BYPASS WAIT |
| | 2511 | DISK_FORMAT PROC NEAR | | 2498 | POP CX |
| | 2512 | OR MOTOR_STATUS,B0H | | 2499 | ----- DO THE SEEK OPERATION |
| | 2513 | MOV AL,040H | | 2500 | CALL SEEK |
| | 2514 | CALL DPA_SETUP | | 2501 | POP AX ; MOVE TO CORRECT TRACK |
| | 2515 | MOV AH,040H | | 2502 | MOV BH,AX ; RECOVER COMMAND |
| | 2516 | DISK_WRITE PROC NEAR | | 2503 | MOV DH,AX ; SAVE COMMAND IN BH |
| | 2517 | OR MOTOR_STATUS,B0H | | 2504 | MOV DH,0 ; SET NO SECTORS READ IN CASE OF ERROR |
| | 2518 | CALL DPA_SETUP | | 2505 | JC J17 ; IF ERROR, THEN EXIT AFTER MOTOR OFF |
| | 2519 | MOV AH,040H | | 2506 | MOV SI,OFFSET J17 ; DUPLY RETURN ON STACK FOR NEC_OUTPUT |
| | 2520 | DISK_WRITE ENDP | | 2507 | PUSH SI ; SO THAT IT WILL RETURN TO MOTOR OFF LOCATION |
| | 2521 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2508 | ----- SEND OUT THE PARAMETERS TO THE CONTROLLER |
| | 2522 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2509 | CALL NEC_OUTPUT |
| | 2523 | RH_OPEN PROC NEAR | | 2510 | MOV AH,1BP+11 ; GET THE CURRENT HEAD NUMBER |
| | 2524 | MOV AL,0 | | 2511 | SAL AH,1 ; MOVE IT TO BIT 2 |
| | 2525 | RET | | 2512 | SAL AH,1 ; ISOLATE THAT BIT |
| | 2526 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2513 | AND AH,4 ; ON IN THE DRIVE NUMBER |
| | 2527 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2514 | CALL NEC_OUTPUT |
| | 2528 | RH_OPEN PROC NEAR | | 2515 | ----- TEST FOR FORMAT COMMAND |
| | 2529 | MOV AL,0 | | 2516 | CHP BH,040H ; IS THIS A FORMAT OPERATION |
| | 2530 | RET | | 2517 | J1E J15 ; NO, CONTINUE WITH R/W/V |
| | 2531 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2518 | JMP J10 ; IF SO, HANDLE SPECIAL |
| | 2532 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2519 | ----- CYLINDER NUMBER |
| | 2533 | RH_OPEN PROC NEAR | | 2520 | MOV AH,CH |
| | 2534 | MOV AL,0 | | 2521 | CALL NEC_OUTPUT |
| | 2535 | RET | | 2522 | MOV AH,1BP+11 ; HEAD NUMBER FROM STACK |
| | 2536 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2523 | CALL NEC_OUTPUT |
| | 2537 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2524 | MOV AH,CL ; SECTOR NUMBER |
| | 2538 | RH_OPEN PROC NEAR | | 2525 | CALL NEC_OUTPUT |
| | 2539 | MOV AL,0 | | 2526 | MOV BX,7 ; BYTES/SECTOR PARM FROM BLOCK |
| | 2540 | RET | | 2527 | CALL GET_PARM ; TO THE NEC |
| | 2541 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2528 | MOV BX,9 ; EOT PARM FROM BLOCK |
| | 2542 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2529 | CALL GET_PARM ; TO THE NEC |
| | 2543 | RH_OPEN PROC NEAR | | 2530 | MOV BX,11 ; GAP LENGTH PARM FROM BLOCK |
| | 2544 | MOV AL,0 | | 2531 | CALL GET_PARM ; TO THE NEC |
| | 2545 | RET | | 2532 | MOV BX,13 ; DTL PARM FROM BLOCK |
| | 2546 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2533 | CALL GET_PARM ; TO THE NEC |
| | 2547 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2534 | MOV SI ; CAN NOW DISCARD THAT DUMMY RETURN ADDRESS |
| | 2548 | RH_OPEN PROC NEAR | | 2535 | ----- LET THE OPERATION HAPPEN |
| | 2549 | MOV AL,0 | | 2536 | CALL WAIT_INT ; WAIT FOR THE INTERRUPT |
| | 2550 | RET | | 2537 | MOV SI ; MOTOR_OFF |
| | 2551 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2538 | J21 J21 ; LOOK FOR ERROR |
| | 2552 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2539 | CALL RESULTS ; GET THE NEC STATUS |
| | 2553 | RH_OPEN PROC NEAR | | 2540 | JC J20 ; LOOK FOR ERROR |
| | 2554 | MOV AL,0 | | 2541 | ----- CHECK THE RESULTS RETURNED BY THE CONTROLLER |
| | 2555 | RET | | 2542 | CID ; SET THE CORRECT DIRECTION |
| | 2556 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2543 | MOV SI,OFFSET NEC_STATUS ; POINT TO STATUS FIELD |
| | 2557 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2544 | MOV SI ; GET SI |
| | 2558 | RH_OPEN PROC NEAR | | 2545 | AND AL,00H ; TEST FOR NORMAL TERMINATION |
| | 2559 | MOV AL,0 | | 2546 | EDF 24C0 |
| | 2560 | RET | | 2547 | EDF 24C0 |
| | 2561 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2548 | EDF 24C0 |
| | 2562 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2549 | EDF 24C0 |
| | 2563 | RH_OPEN PROC NEAR | | 2550 | EDF 24C0 |
| | 2564 | MOV AL,0 | | 2551 | EDF 24C0 |
| | 2565 | RET | | 2552 | EDF 24C0 |
| | 2566 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2553 | EDF 24C0 |
| | 2567 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2554 | EDF 24C0 |
| | 2568 | RH_OPEN PROC NEAR | | 2555 | EDF 24C0 |
| | 2569 | MOV AL,0 | | 2556 | EDF 24C0 |
| | 2570 | RET | | 2557 | EDF 24C0 |
| | 2571 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2558 | EDF 24C0 |
| | 2572 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2559 | EDF 24C0 |
| | 2573 | RH_OPEN PROC NEAR | | 2560 | EDF 24C0 |
| | 2574 | MOV AL,0 | | 2561 | EDF 24C0 |
| | 2575 | RET | | 2562 | EDF 24C0 |
| | 2576 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2563 | EDF 24C0 |
| | 2577 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2564 | EDF 24C0 |
| | 2578 | RH_OPEN PROC NEAR | | 2565 | EDF 24C0 |
| | 2579 | MOV AL,0 | | 2566 | EDF 24C0 |
| | 2580 | RET | | 2567 | EDF 24C0 |
| | 2581 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2568 | EDF 24C0 |
| | 2582 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2569 | EDF 24C0 |
| | 2583 | RH_OPEN PROC NEAR | | 2570 | EDF 24C0 |
| | 2584 | MOV AL,0 | | 2571 | EDF 24C0 |
| | 2585 | RET | | 2572 | EDF 24C0 |
| | 2586 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2573 | EDF 24C0 |
| | 2587 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2574 | EDF 24C0 |
| | 2588 | RH_OPEN PROC NEAR | | 2575 | EDF 24C0 |
| | 2589 | MOV AL,0 | | 2576 | EDF 24C0 |
| | 2590 | RET | | 2577 | EDF 24C0 |
| | 2591 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2578 | EDF 24C0 |
| | 2592 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2579 | EDF 24C0 |
| | 2593 | RH_OPEN PROC NEAR | | 2580 | EDF 24C0 |
| | 2594 | MOV AL,0 | | 2581 | EDF 24C0 |
| | 2595 | RET | | 2582 | EDF 24C0 |
| | 2596 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2583 | EDF 24C0 |
| | 2597 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2584 | EDF 24C0 |
| | 2598 | RH_OPEN PROC NEAR | | 2585 | EDF 24C0 |
| | 2599 | MOV AL,0 | | 2586 | EDF 24C0 |
| | 2600 | RET | | 2587 | EDF 24C0 |
| | 2601 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2588 | EDF 24C0 |
| | 2602 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2589 | EDF 24C0 |
| | 2603 | RH_OPEN PROC NEAR | | 2590 | EDF 24C0 |
| | 2604 | MOV AL,0 | | 2591 | EDF 24C0 |
| | 2605 | RET | | 2592 | EDF 24C0 |
| | 2606 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2593 | EDF 24C0 |
| | 2607 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2594 | EDF 24C0 |
| | 2608 | RH_OPEN PROC NEAR | | 2595 | EDF 24C0 |
| | 2609 | MOV AL,0 | | 2596 | EDF 24C0 |
| | 2610 | RET | | 2597 | EDF 24C0 |
| | 2611 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2598 | EDF 24C0 |
| | 2612 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2599 | EDF 24C0 |
| | 2613 | RH_OPEN PROC NEAR | | 2600 | EDF 24C0 |
| | 2614 | MOV AL,0 | | 2601 | EDF 24C0 |
| | 2615 | RET | | 2602 | EDF 24C0 |
| | 2616 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2603 | EDF 24C0 |
| | 2617 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2604 | EDF 24C0 |
| | 2618 | RH_OPEN PROC NEAR | | 2605 | EDF 24C0 |
| | 2619 | MOV AL,0 | | 2606 | EDF 24C0 |
| | 2620 | RET | | 2607 | EDF 24C0 |
| | 2621 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2608 | EDF 24C0 |
| | 2622 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2609 | EDF 24C0 |
| | 2623 | RH_OPEN PROC NEAR | | 2610 | EDF 24C0 |
| | 2624 | MOV AL,0 | | 2611 | EDF 24C0 |
| | 2625 | RET | | 2612 | EDF 24C0 |
| | 2626 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2613 | EDF 24C0 |
| | 2627 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2614 | EDF 24C0 |
| | 2628 | RH_OPEN PROC NEAR | | 2615 | EDF 24C0 |
| | 2629 | MOV AL,0 | | 2616 | EDF 24C0 |
| | 2630 | RET | | 2617 | EDF 24C0 |
| | 2631 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2618 | EDF 24C0 |
| | 2632 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2619 | EDF 24C0 |
| | 2633 | RH_OPEN PROC NEAR | | 2620 | EDF 24C0 |
| | 2634 | MOV AL,0 | | 2621 | EDF 24C0 |
| | 2635 | RET | | 2622 | EDF 24C0 |
| | 2636 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2623 | EDF 24C0 |
| | 2637 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2624 | EDF 24C0 |
| | 2638 | RH_OPEN PROC NEAR | | 2625 | EDF 24C0 |
| | 2639 | MOV AL,0 | | 2626 | EDF 24C0 |
| | 2640 | RET | | 2627 | EDF 24C0 |
| | 2641 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2628 | EDF 24C0 |
| | 2642 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2629 | EDF 24C0 |
| | 2643 | RH_OPEN PROC NEAR | | 2630 | EDF 24C0 |
| | 2644 | MOV AL,0 | | 2631 | EDF 24C0 |
| | 2645 | RET | | 2632 | EDF 24C0 |
| | 2646 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2633 | EDF 24C0 |
| | 2647 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2634 | EDF 24C0 |
| | 2648 | RH_OPEN PROC NEAR | | 2635 | EDF 24C0 |
| | 2649 | MOV AL,0 | | 2636 | EDF 24C0 |
| | 2650 | RET | | 2637 | EDF 24C0 |
| | 2651 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2638 | EDF 24C0 |
| | 2652 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2639 | EDF 24C0 |
| | 2653 | RH_OPEN PROC NEAR | | 2640 | EDF 24C0 |
| | 2654 | MOV AL,0 | | 2641 | EDF 24C0 |
| | 2655 | RET | | 2642 | EDF 24C0 |
| | 2656 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2643 | EDF 24C0 |
| | 2657 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2644 | EDF 24C0 |
| | 2658 | RH_OPEN PROC NEAR | | 2645 | EDF 24C0 |
| | 2659 | MOV AL,0 | | 2646 | EDF 24C0 |
| | 2660 | RET | | 2647 | EDF 24C0 |
| | 2661 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2648 | EDF 24C0 |
| | 2662 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2649 | EDF 24C0 |
| | 2663 | RH_OPEN PROC NEAR | | 2650 | EDF 24C0 |
| | 2664 | MOV AL,0 | | 2651 | EDF 24C0 |
| | 2665 | RET | | 2652 | EDF 24C0 |
| | 2666 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2653 | EDF 24C0 |
| | 2667 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2654 | EDF 24C0 |
| | 2668 | RH_OPEN PROC NEAR | | 2655 | EDF 24C0 |
| | 2669 | MOV AL,0 | | 2656 | EDF 24C0 |
| | 2670 | RET | | 2657 | EDF 24C0 |
| | 2671 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2658 | EDF 24C0 |
| | 2672 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2659 | EDF 24C0 |
| | 2673 | RH_OPEN PROC NEAR | | 2660 | EDF 24C0 |
| | 2674 | MOV AL,0 | | 2661 | EDF 24C0 |
| | 2675 | RET | | 2662 | EDF 24C0 |
| | 2676 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2663 | EDF 24C0 |
| | 2677 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2664 | EDF 24C0 |
| | 2678 | RH_OPEN PROC NEAR | | 2665 | EDF 24C0 |
| | 2679 | MOV AL,0 | | 2666 | EDF 24C0 |
| | 2680 | RET | | 2667 | EDF 24C0 |
| | 2681 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2668 | EDF 24C0 |
| | 2682 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2669 | EDF 24C0 |
| | 2683 | RH_OPEN PROC NEAR | | 2670 | EDF 24C0 |
| | 2684 | MOV AL,0 | | 2671 | EDF 24C0 |
| | 2685 | RET | | 2672 | EDF 24C0 |
| | 2686 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2673 | EDF 24C0 |
| | 2687 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2674 | EDF 24C0 |
| | 2688 | RH_OPEN PROC NEAR | | 2675 | EDF 24C0 |
| | 2689 | MOV AL,0 | | 2676 | EDF 24C0 |
| | 2690 | RET | | 2677 | EDF 24C0 |
| | 2691 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2678 | EDF 24C0 |
| | 2692 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2679 | EDF 24C0 |
| | 2693 | RH_OPEN PROC NEAR | | 2680 | EDF 24C0 |
| | 2694 | MOV AL,0 | | 2681 | EDF 24C0 |
| | 2695 | RET | | 2682 | EDF 24C0 |
| | 2696 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2683 | EDF 24C0 |
| | 2697 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2684 | EDF 24C0 |
| | 2698 | RH_OPEN PROC NEAR | | 2685 | EDF 24C0 |
| | 2699 | MOV AL,0 | | 2686 | EDF 24C0 |
| | 2700 | RET | | 2687 | EDF 24C0 |
| | 2701 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2688 | EDF 24C0 |
| | 2702 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2689 | EDF 24C0 |
| | 2703 | RH_OPEN PROC NEAR | | 2690 | EDF 24C0 |
| | 2704 | MOV AL,0 | | 2691 | EDF 24C0 |
| | 2705 | RET | | 2692 | EDF 24C0 |
| | 2706 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2693 | EDF 24C0 |
| | 2707 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2694 | EDF 24C0 |
| | 2708 | RH_OPEN PROC NEAR | | 2695 | EDF 24C0 |
| | 2709 | MOV AL,0 | | 2696 | EDF 24C0 |
| | 2710 | RET | | 2697 | EDF 24C0 |
| | 2711 | DISK_WRITE ROUTINE TO FALL INTO RH_OPEN | | 2698 | EDF 24C0 |
| | 2712 | THIS ROUTINE PERFORMS THE READ/WRITE/VERIFY OPERATION | | 2699 | EDF 24C0 |
| | 2713 | RH_OPEN PROC NEAR | | 2700 | EDF 24C0 |
| | 2714 | MOV AL,0 | | 2701 | EDF 24C0 |
| | 2715 | RET | | 2702 | EDF 24C0 |
| | 2716</ | | | | |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|---------------|------|---|-----------------|------|---------------------|
| EE01 743B | 2561 | JZ J22 | EE52 800E410080 | 2639 | OR |
| EE03 3C40 | 2562 | CHP AL-00H | EE57 59 | 2640 | POP CX |
| EE05 7829 | 2563 | JNZ J16 | EE58 5A | 2641 | POP DX |
| | 2564 | | EE59 5B | 2642 | POP AX |
| | 2565 | ----- ABNORMAL TERMINATION, FIND OUT WHY | EE5A F9 | 2643 | STC |
| EE07 AC | 2566 | LODS REC_STATUS | EE5B C3 | 2644 | RET |
| EE08 D0E0 | 2567 | SAL AL-1 | EE5C | 2645 | |
| EE0A B404 | 2568 | MOV AH-RECORD_NOT_FND | EE5C 33C9 | 2646 | XOR CX,CX |
| EE0C 7824 | 2569 | JC J19 | EE5E | 2647 | |
| EE0E D0E0 | 2570 | SAL AL-1 | EE5E EC | 2648 | IN AL,DX |
| EE10 D0E0 | 2571 | SAL AL-1 | EE5F A800 | 2649 | TEST AL,00H |
| EE12 B410 | 2572 | MOV AH-BAD_CRC | EE61 7504 | 2650 | TEST J27 |
| EE14 721C | 2573 | JC J19 | EE63 E2F9 | 2651 | JNZ J27 |
| EE16 D0E0 | 2574 | SAL AL-1 | EE65 E8E8 | 2652 | LOD J26 |
| EE18 B408 | 2575 | MOV AH-BAD_OHA | EE67 | 2653 | JMP J24 |
| EE1A 7216 | 2576 | JC J19 | EE67 | 2654 | |
| EE1C D0E0 | 2577 | SAL AL-1 | EE67 B4C4 | 2655 | MOV AL,AH |
| EE1E D0E0 | 2578 | SAL AL-1 | EE69 B4F503 | 2656 | MOV DX,07F5H |
| EE20 B404 | 2579 | MOV AH-RECORD_NOT_FND | EE6C EE | 2657 | OUT DX,AL |
| EE22 720E | 2580 | JC J19 | EE6D 59 | 2658 | POP CX |
| EE24 D0E0 | 2581 | SAL AL-1 | EE6E 5A | 2659 | POP DX |
| EE26 B403 | 2582 | MOV AH-WRITE_PROTECT | EE6F C3 | 2660 | |
| EE28 7208 | 2583 | JC J19 | | 2661 | RET |
| EE2A D0E0 | 2584 | SAL AL-1 | | 2662 | NEC_OUTPUT |
| EE2C B402 | 2585 | MOV AH-BAD_ADDR_MARK | | 2663 | GET_PARM |
| EE2E 7202 | 2586 | JC J19 | | 2664 | GET_PARM |
| | 2587 | ----- NEC MUST HAVE FAILED | | 2665 | GET_PARM |
| EE30 | 2588 | | | 2666 | GET_PARM |
| EE30 B420 | 2589 | J16: MOV AH-BAD_NEC | EE70 | 2676 | PROC NEAR |
| EE32 | 2590 | | EE70 1E | 2677 | PUSH DS |
| EE32 00264100 | 2591 | J19: OR DISKETTE_STATUS,AH | EE71 28C0 | 2678 | SUB AX,AX |
| EE36 E87701 | 2592 | CALL NUM_TRANS | EE73 E8D8 | 2679 | MOV DS,AX |
| EE39 C3 | 2593 | RET | EE75 C5367800 | 2680 | ASSUME DS:ABS0 |
| | 2594 | | EE79 D1E8 | 2681 | LDS SI,DISK_POINTER |
| | 2595 | | EE7B 8420 | 2682 | SHR BX-1 |
| | 2596 | | EE7D 1F | 2683 | MOV AH-151-BX1 |
| | 2597 | | EE7E 72C4 | 2684 | POP DS |
| EE3A | 2598 | J21: CALL RESULTS | EE7F 72C4 | 2685 | ASSUME DS:DATA |
| EE3A E80101 | 2599 | RET | EE80 C3 | 2686 | JC NEC_OUTPUT |
| EE3D C3 | 2600 | | | 2687 | RET |
| | 2601 | | | 2688 | GET_PARM |
| | 2602 | ----- OPERATION WAS SUCCESSFUL | | 2689 | GET_PARM |
| EE3E | 2603 | | | 2690 | GET_PARM |
| EE3E E80701 | 2604 | J22: CALL NUM_TRANS | | 2691 | GET_PARM |
| EE41 32E4 | 2605 | XOR AH,AH | | 2692 | GET_PARM |
| EE43 C3 | 2606 | RET | | 2693 | GET_PARM |
| | 2607 | | | 2694 | GET_PARM |
| | 2608 | | | 2695 | GET_PARM |
| | 2609 | | | 2696 | GET_PARM |
| | 2610 | | | 2697 | GET_PARM |
| | 2611 | ----- NEC_OUTPUT | | 2698 | GET_PARM |
| | 2612 | THIS ROUTINE SENDS A BYTE TO THE NEC CONTROLLER | | 2699 | GET_PARM |
| | 2613 | AFTER TESTING FOR CORRECT DIRECTION AND CONTROLLER READ | | 2700 | GET_PARM |
| | 2614 | THIS ROUTINE WILL TIME OUT IF THE BYTE IS NOT ACCEPTED | | 2701 | GET_PARM |
| | 2615 | WITHIN A REASONABLE AMOUNT OF TIME, SETTING THE DISKETTE STATUS | | 2702 | GET_PARM |
| | 2616 | ON COMPLETION | | 2703 | GET_PARM |
| | 2617 | INPUT | | 2704 | GET_PARM |
| | 2618 | (AH) BYTE TO BE OUTPUT | | 2705 | GET_PARM |
| | 2619 | OUTPUT | | 2706 | GET_PARM |
| | 2620 | CT = 0 SUCCESS | | 2707 | GET_PARM |
| | 2621 | IF A FAILURE HAS OCCURRED, THE RETURN IS MADE ONE LEVEL | | 2708 | GET_PARM |
| | 2622 | HIGHER THAN THE CALLER OF NEC_OUTPUT | | 2709 | GET_PARM |
| | 2623 | THIS REMOVES THE REQUIREMENT OF TESTING AFTER EVERY CALL | | 2710 | GET_PARM |
| | 2624 | OF NEC_OUTPUT | | 2711 | GET_PARM |
| | 2625 | CALL DESTROYED | | 2712 | GET_PARM |
| | 2626 | ----- | | 2713 | GET_PARM |
| EE44 52 | 2627 | NEC_OUTPUT PROC NEAR | EE81 B801 | 2714 | CALL NEC_OUTPUT |
| EE45 51 | 2628 | PUSH CX | EE83 51 | 2715 | CALL NEC_OUTPUT |
| EE46 B47403 | 2629 | POP CX | EE84 BACA | | |
| EE49 33C9 | 2630 | MOV DX,03F4H | EE8A D0C0 | | |
| EE4B | 2631 | XOR CX,CX | EE8B 59 | | |
| EE4B EC | 2632 | J23: IN AL,DX | EE89 840A3E00 | | |
| EE4C A840 | 2633 | TEST AL,00H | EE8F 080A3E00 | | |
| EE4E 740C | 2634 | JZ J25 | EE93 B407 | | |
| EE50 E2F9 | 2635 | LOOP | EE95 E8CFF | | |
| EE52 | 2636 | | EE9A E87FF | | |
| | 2637 | | | | |
| | 2638 | | | | |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|---------------|------|--|-----------|------|---|
| EE00 E07200 | 2716 | CALL CHM_STAT_2 | EF02 E605 | 2793 | OUT DMA+5.AL |
| EEA0 7259 | 2717 | JC J32 | EF04 BAC4 | 2794 | MOV AL,AN |
| | 2718 | | EF06 E605 | 2795 | OUT DMA+5.AL |
| | 2719 | | EF08 E605 | 2796 | POP CX |
| | 2720 | | EF09 58 | 2797 | POP AX |
| EEA2 B40F | 2721 | J2B: MOV AN,OFM | EF0A 93C1 | 2798 | ADD AX,CX |
| EEA4 E070FF | 2722 | CALL HEC_OUTPUT | EF0C 59 | 2799 | POP CX |
| EEA7 B4C2 | 2723 | MOV AN,OL | EF0D B002 | 2800 | MOV AL,2 |
| EEA9 E0B0FF | 2724 | CALL HEC_OUTPUT | EF0F B0A4 | 2801 | OUT DMA+10.AL |
| EEAB B4E5 | 2725 | MOV AN,CH | EF11 C3 | 2802 | RET |
| EEAD E0B3FF | 2726 | CALL HEC_OUTPUT | | 2803 | DMA_SETUP |
| EEAE E0B500 | 2727 | CALL CHM_STAT_2 | | 2804 | ENDP |
| EEB1 E0B500 | 2728 | CALL CHM_STAT_2 | | 2805 | CHM_STAT_2 |
| | 2729 | | | 2806 | THIS ROUTINE HANDLES THE INTERRUPT RECEIVED AFTER |
| | 2730 | ----- WAIT FOR HEAD SETTLE | | 2807 | A RECALIBRATE, SEEK, OR RESET TO THE ADAPTER. |
| | 2731 | | | 2808 | THE INTERRUPT IS WAITED FOR, THE INTERRUPT STATUS SENSED, |
| EEB4 9C | 2732 | PUSHF | | 2809 | AND THE RESULT RETURNED TO THE CALLER. |
| EEB5 B01200 | 2733 | MOV BX,16 | | 2810 | INPUT |
| EEB6 E0B5FF | 2734 | CALL GET_PARM | | 2811 | OUTPUT |
| EEB8 51 | 2735 | PUSH CX | | 2812 | CT = 0 SUCCESS |
| EEBC | 2736 | | | 2813 | CT = 1 FAILURE -- ERROR IN DISKETTE STATUS |
| EEBC B97602 | 2737 | MOV CX,550 | | 2814 | (AX) DESTROYED |
| EEBF 0A54 | 2738 | OR AN,AN | | 2815 | ----- |
| EEC1 7456 | 2739 | JZ J31 | | 2816 | CHM_STAT_2 PROC NEAR |
| EEC3 E7FE | 2740 | LOOP J30 | | 2817 | CALL WAIT_INT |
| EEC5 FECC | 2741 | DEC AN | | 2818 | JC J34 |
| EEC7 E0F3 | 2742 | JMP J29 | | 2819 | MOV AN,OFM |
| EEC9 | 2743 | | | 2820 | CALL HEC_OUTPUT |
| EECA 90 | 2744 | POP CX | | 2821 | CALL RESULTS |
| EECB 90 | 2745 | POPFF | | 2822 | CALL JC J34 |
| EECD C3 | 2746 | RET | | 2823 | JC J34 |
| | 2747 | | | 2824 | MOV AL,HEC_STATUS |
| | 2748 | | | 2825 | AND AL,OFM |
| | 2749 | | | 2826 | CHP AL,OFM |
| | 2750 | DMA_SETUP | | 2827 | JZ J35 |
| | 2751 | THIS ROUTINE SETS UP THE DMA FOR READ/WRITE/VERIFY | | 2828 | CLC |
| | 2752 | OPERATIONS. | | 2829 | J34: RET |
| | 2753 | INPUT | | 2830 | J35: OR DISKETTE_STATUS,BAD_SEEK |
| | 2754 | (AL) = MODE BYTE FOR THE DMA | | 2831 | STC |
| | 2755 | (ES:BX) = ADDRESS TO READ/WRITE THE DATA | | 2832 | REF |
| | 2756 | OUTPUT | | 2833 | CHM_STAT_2 ENDP |
| | 2757 | (AX) DESTROYED | | 2834 | ----- |
| | 2758 | | | 2835 | WAIT_INT |
| DMA_SETUP | 2759 | PROC NEAR | | 2836 | THIS ROUTINE WAITS FOR AN INTERRUPT TO OCCUR |
| PUSH CX | 2760 | | | 2837 | AT THE OUT ROUTINE TAKES PLACE OUTSIDE THE WAIT, SO |
| OUT DMA+12.AL | 2761 | | | 2838 | THAT AN ERROR MAY BE RETURNED IF THE DRIVE IS NOT READY |
| OUT DMA+11.AL | 2762 | | | 2839 | INPUT |
| MOV AX,ES | 2763 | | | 2840 | NONE |
| MOV CL,4 | 2764 | | | 2841 | OUTPUT |
| ROL AX,CL | 2765 | | | 2842 | CT = 0 SUCCESS |
| MOV CH,AL | 2766 | | | 2843 | CT = 1 FAILURE -- DISKETTE_STATUS IS SET ACCORDINGLY |
| AND AN,OFM | 2767 | | | 2844 | (AX) DESTROYED |
| ADD AX,BX | 2768 | | | 2845 | ----- |
| JNC J33 | 2769 | | | 2846 | WAIT_INT PROC NEAR |
| JNC CN | 2770 | | | 2847 | TURN ON INTERRUPTS, JUST IN CASE |
| J33: PUSH AX | 2771 | | | 2848 | STI |
| OUT DMA+4.AL | 2772 | | | 2849 | PUSH BX |
| MOV AL,AN | 2773 | | | 2850 | SAVE REGISTERS |
| OUT DMA+4.AL | 2774 | | | 2851 | PUSH CX |
| MOV AL,CH | 2775 | | | 2852 | CLEAR THE COUNTERS |
| AND AN,OFM | 2776 | | | 2853 | MOV BL,2 |
| OUT 0B1H.AL | 2777 | | | 2854 | FOR 2 SECOND WAIT |
| OUT | 2778 | | | 2855 | XOR CX,CX |
| | 2779 | ----- DETERMINE COUNT | | 2856 | J36: TEST SEEK_STATUS.INT_FLAG |
| | 2780 | | | 2857 | J37 |
| | 2781 | | | 2858 | J36 |
| | 2782 | | | 2859 | LOOP J36 |
| | 2783 | | | 2860 | DEC BL |
| | 2784 | | | 2861 | J36 |
| | 2785 | | | 2862 | OR DISKETTE_STATUS,TIME_OUT |
| | 2786 | | | 2863 | IF NOTHING HAPPENED |
| | 2787 | | | 2864 | ERROR RETURN |
| | 2788 | | | 2865 | STC |
| | 2789 | | | 2866 | J37: PUSHF |
| | 2790 | | | 2867 | AND |
| | 2791 | | | 2868 | SEEK_STATUS,NOT_INT_FLAG |
| | 2792 | | | 2869 | TURN OFF INTERRUPT FLAG |
| | | | | 2870 | RECOVER CARRY |
| | | | | 2871 | POP CX |
| | | | | 2872 | POP BX |
| | | | | 2873 | RECOVER REGISTERS |
| | | | | 2874 | GOOD RETURN CODE COMES FROM TEST INST |
| | | | | 2875 | ENDP |
| | | | | 2876 | WAIT_INT |
| | | | | 2877 | |
| | | | | 2878 | |
| | | | | 2879 | |
| | | | | 2880 | |
| | | | | 2881 | |
| | | | | 2882 | |
| | | | | 2883 | |
| | | | | 2884 | |
| | | | | 2885 | |
| | | | | 2886 | |
| | | | | 2887 | |
| | | | | 2888 | |
| | | | | 2889 | |
| | | | | 2890 | |
| | | | | 2891 | |
| | | | | 2892 | |
| | | | | 2893 | |
| | | | | 2894 | |
| | | | | 2895 | |
| | | | | 2896 | |
| | | | | 2897 | |
| | | | | 2898 | |
| | | | | 2899 | |
| | | | | 2900 | |
| | | | | 2901 | |
| | | | | 2902 | |
| | | | | 2903 | |
| | | | | 2904 | |
| | | | | 2905 | |
| | | | | 2906 | |
| | | | | 2907 | |
| | | | | 2908 | |
| | | | | 2909 | |
| | | | | 2910 | |
| | | | | 2911 | |
| | | | | 2912 | |
| | | | | 2913 | |
| | | | | 2914 | |
| | | | | 2915 | |
| | | | | 2916 | |
| | | | | 2917 | |
| | | | | 2918 | |
| | | | | 2919 | |
| | | | | 2920 | |
| | | | | 2921 | |
| | | | | 2922 | |
| | | | | 2923 | |
| | | | | 2924 | |
| | | | | 2925 | |
| | | | | 2926 | |
| | | | | 2927 | |
| | | | | 2928 | |
| | | | | 2929 | |
| | | | | 2930 | |
| | | | | 2931 | |
| | | | | 2932 | |
| | | | | 2933 | |
| | | | | 2934 | |
| | | | | 2935 | |
| | | | | 2936 | |
| | | | | 2937 | |
| | | | | 2938 | |
| | | | | 2939 | |
| | | | | 2940 | |
| | | | | 2941 | |
| | | | | 2942 | |
| | | | | 2943 | |
| | | | | 2944 | |
| | | | | 2945 | |
| | | | | 2946 | |
| | | | | 2947 | |
| | | | | 2948 | |
| | | | | 2949 | |
| | | | | 2950 | |
| | | | | 2951 | |
| | | | | 2952 | |
| | | | | 2953 | |
| | | | | 2954 | |
| | | | | 2955 | |
| | | | | 2956 | |
| | | | | 2957 | |
| | | | | 2958 | |
| | | | | 2959 | |
| | | | | 2960 | |
| | | | | 2961 | |
| | | | | 2962 | |
| | | | | 2963 | |
| | | | | 2964 | |
| | | | | 2965 | |
| | | | | 2966 | |
| | | | | 2967 | |
| | | | | 2968 | |
| | | | | 2969 | |
| | | | | 2970 | |
| | | | | 2971 | |
| | | | | 2972 | |

[illegible]

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|---------|------|--|---------|------|--------|
| | 3007 | INT 17 | | 3007 | |
| | 3008 | PRINTING | | 3008 | |
| | 3009 | THIS ROUTINE PROVIDES COMMUNICATION WITH THE PRINTER | | 3009 | |
| | 3010 | (AH)=0 PRINT THE CHARACTER IN (AL) | | 3010 | |
| | 3011 | ON RETURN, AH=1 IF CHARACTER COULD NOT BE PRINTED (TIME OUT) | | 3011 | |
| | 3012 | OTHER BITS SET AS ON NORMAL STATUS CALL | | 3012 | |
| | 3013 | (AH)=1 INITIALIZE THE PRINTER PORT | | 3013 | |
| | 3014 | REPLUS WITH (AH) SET WITH PRINTER STATUS | | 3014 | |
| | 3015 | (AH)=2 READ THE PRINTER STATUS INTO (AH) | | 3015 | |
| | 3016 | | | 3016 | |
| | 3017 | | | 3017 | |
| | 3018 | | | 3018 | |
| | 3019 | | | 3019 | |
| | 3020 | | | 3020 | |
| | 3021 | | | 3021 | |
| | 3022 | | | 3022 | |
| | 3023 | | | 3023 | |
| | 3024 | | | 3024 | |
| | 3025 | (DX)=1 PRINTER TO BE USED (0,1,2) CORRESPONDING TO ACTUAL VALUES | | 3025 | |
| | 3026 | IN PRINTER BASE AREA | | 3026 | |
| | 3027 | DATA AREA PRINTER_BASE CONTAINS THE BASE ADDRESS OF THE PRINTER CARD(S) | | 3027 | |
| | 3028 | AVAILABLE (LOCATED AT BEGINNING OF DATA SEGMENT, 400H ABSOLUTE, 3 WORDS) | | 3028 | |
| | 3029 | REGISTERS | | 3029 | |
| | 3030 | AH IS MODIFIED | | 3030 | |
| | 3031 | ALL OTHERS UNCHANGED | | 3031 | |
| | 3032 | ASSUME CS:CODE,DS:DATA | | 3032 | |
| | 3033 | PRINTING | | 3033 | |
| | 3034 | INTERRUPTS BACK ON | | 3034 | |
| | 3035 | SAVE SEGMENT | | 3035 | |
| | 3036 | PUSH DS | | 3036 | |
| | 3037 | PUSH DX | | 3037 | |
| | 3038 | PUSH SI | | 3038 | |
| | 3039 | PUSH CX | | 3039 | |
| | 3040 | PUSH BX | | 3040 | |
| | 3041 | MOV SI,DATA | | 3041 | |
| | 3042 | MOV DS,SI | | 3042 | |
| | 3043 | MOV SI,DX | | 3043 | |
| | 3044 | SHL SI,1 | | 3044 | |
| | 3045 | MOV DX,PRINTER_BASE[SI] | | 3045 | |
| | 3046 | OR DX,DX | | 3046 | |
| | 3047 | JZ B1 | | 3047 | |
| | 3048 | OR AH,AH | | 3048 | |
| | 3049 | JZ B2 | | 3049 | |
| | 3050 | DEC AH | | 3050 | |
| | 3051 | JZ B3 | | 3051 | |
| | 3052 | DEC AH | | 3052 | |
| | 3053 | JZ B5 | | 3053 | |
| | 3054 | POP BX | | 3054 | |
| | 3055 | POP CX | | 3055 | |
| | 3056 | POP SI | | 3056 | |
| | 3057 | POP DX | | 3057 | |
| | 3058 | POP DS | | 3058 | |
| | 3059 | IRET | | 3059 | |
| | 3060 | | | 3060 | |
| | 3061 | PRINT THE CHARACTER IN (AL) | | 3061 | |
| | 3062 | | | 3062 | |
| | 3063 | PUSH AX | | 3063 | |
| | 3064 | MOV BL,10 | | 3064 | |
| | 3065 | XOR CX,CX | | 3065 | |
| | 3066 | OUT DX,AL | | 3066 | |
| | 3067 | INC DX | | 3067 | |
| | 3068 | JNC B1 | | 3068 | |
| | 3069 | MOV AL,DX | | 3069 | |
| | 3070 | MOV AH,AL | | 3070 | |
| | 3071 | TEST AL,BN | | 3071 | |
| | 3072 | JNZ B4 | | 3072 | |
| | 3073 | LOOP B3 | | 3073 | |
| | 3074 | DEC BL | | 3074 | |
| | 3075 | JNZ B3 | | 3075 | |
| | 3076 | OR AH,1 | | 3076 | |
| | 3077 | MOV AH,0FH | | 3077 | |
| | 3078 | JMP SHORT B7 | | 3078 | |
| | 3079 | MOV AL,0DH | | 3079 | |
| | 3080 | JNC B4 | | 3080 | |
| | 3081 | JNC B4 | | 3081 | |
| | 3082 | JNC B4 | | 3082 | |
| | 3083 | JNC B4 | | 3083 | |
| | 3084 | JNC B4 | | 3084 | |
| | 3085 | JNC B4 | | 3085 | |
| | 3086 | JNC B4 | | 3086 | |
| | 3087 | JNC B4 | | 3087 | |
| | 3088 | JNC B4 | | 3088 | |
| | 3089 | JNC B4 | | 3089 | |
| | 3090 | JNC B4 | | 3090 | |
| | 3091 | JNC B4 | | 3091 | |
| | 3092 | JNC B4 | | 3092 | |
| | 3093 | JNC B4 | | 3093 | |
| | 3094 | JNC B4 | | 3094 | |
| | 3095 | JNC B4 | | 3095 | |
| | 3096 | JNC B4 | | 3096 | |
| | 3097 | JNC B4 | | 3097 | |
| | 3098 | JNC B4 | | 3098 | |
| | 3099 | JNC B4 | | 3099 | |
| | 3100 | JNC B4 | | 3100 | |
| | 3101 | JNC B4 | | 3101 | |
| | 3102 | JNC B4 | | 3102 | |
| | 3103 | JNC B4 | | 3103 | |
| | 3104 | JNC B4 | | 3104 | |
| | 3105 | JNC B4 | | 3105 | |
| | 3106 | JNC B4 | | 3106 | |
| | 3107 | JNC B4 | | 3107 | |
| | 3108 | JNC B4 | | 3108 | |
| | 3109 | JNC B4 | | 3109 | |
| | 3110 | JNC B4 | | 3110 | |
| | 3111 | JNC B4 | | 3111 | |
| | 3112 | JNC B4 | | 3112 | |
| | 3113 | JNC B4 | | 3113 | |
| | 3114 | JNC B4 | | 3114 | |
| | 3115 | JNC B4 | | 3115 | |
| | 3116 | JNC B4 | | 3116 | |
| | 3117 | JNC B4 | | 3117 | |
| | 3118 | JNC B4 | | 3118 | |
| | 3119 | JNC B4 | | 3119 | |
| | 3120 | JNC B4 | | 3120 | |
| | 3121 | JNC B4 | | 3121 | |
| | 3122 | JNC B4 | | 3122 | |
| | 3123 | JNC B4 | | 3123 | |
| | 3124 | JNC B4 | | 3124 | |
| | 3125 | JNC B4 | | 3125 | |
| | 3126 | JNC B4 | | 3126 | |
| | 3127 | JNC B4 | | 3127 | |
| | 3128 | JNC B4 | | 3128 | |
| | 3129 | JNC B4 | | 3129 | |
| | 3130 | JNC B4 | | 3130 | |
| | 3131 | JNC B4 | | 3131 | |
| | 3132 | JNC B4 | | 3132 | |
| | 3133 | JNC B4 | | 3133 | |
| | 3134 | JNC B4 | | 3134 | |
| | 3135 | JNC B4 | | 3135 | |
| | 3136 | JNC B4 | | 3136 | |
| | 3137 | JNC B4 | | 3137 | |
| | 3138 | JNC B4 | | 3138 | |
| | 3139 | JNC B4 | | 3139 | |
| | 3140 | JNC B4 | | 3140 | |
| | 3141 | JNC B4 | | 3141 | |
| | 3142 | JNC B4 | | 3142 | |
| | 3143 | JNC B4 | | 3143 | |
| | 3144 | JNC B4 | | 3144 | |
| | 3145 | JNC B4 | | 3145 | |
| | 3146 | JNC B4 | | 3146 | |
| | 3147 | JNC B4 | | 3147 | |
| | 3148 | JNC B4 | | 3148 | |
| | 3149 | JNC B4 | | 3149 | |
| | 3150 | JNC B4 | | 3150 | |
| | 3151 | JNC B4 | | 3151 | |
| | 3152 | JNC B4 | | 3152 | |
| | 3153 | JNC B4 | | 3153 | |
| | 3154 | JNC B4 | | 3154 | |
| | 3155 | JNC B4 | | 3155 | |
| | 3156 | JNC B4 | | 3156 | |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|---------|------|--|-----------------|------|--|
| | 3157 | (AH)=6 SCROLL ACTIVE PAGE UP | F0A5 | 3243 | MIL |
| | 3158 | (AL) = NUMBER OF LINES, INPUT LINES BLANKED AT BOTTOM OF WINDOW | F0A5 F0A0 | 3244 | DM |
| | 3159 | (CH-CL) = ROW,COLUMN OF UPPER LEFT CORNER OF SCROLL | F0A7 CFF1 | 3245 | DM |
| | 3160 | (CH-OL) = ROW,COLUMN OF LOWER RIGHT CORNER OF SCROLL | F0A9 F0F1 | 3246 | DM |
| | 3161 | (BH) = ATTRIBUTE TO BE USED ON BLANK LINE | F0A8 1A02 | 3247 | DM |
| | 3162 | (AH)=7 SCROLL ACTIVE PAGE DOWN | F0A0 A9F7 | 3248 | DM |
| | 3163 | (AL) = NUMBER OF LINES, INPUT LINES BLANKED AT TOP OF WINDOW | F0A7 30F2 | 3249 | DM |
| | 3164 | (CH-CL) = ROW,COLUMN OF UPPER LEFT CORNER OF SCROLL | F053 61F3 | 3250 | DM |
| | 3165 | (CH-OL) = ROW,COLUMN OF LOWER RIGHT CORNER OF SCROLL | F055 70F3 | 3251 | DM |
| | 3166 | (BH) = ATTRIBUTE TO BE USED ON BLANK LINE | F057 C3F3 | 3252 | DM |
| | 3167 | | F059 F0F3 | 3253 | DM |
| | 3168 | | F05B 34F2 | 3254 | DM |
| | 3169 | | F05D 30F4 | 3255 | DM |
| | 3170 | CHARACTER HANDLING ROUTINES | F05F 27F4 | 3256 | DM |
| | 3171 | (AH) = 8 READ ATTRIBUTE/CHARACTER AT CURRENT CURSOR POSITION | F061 22F7 | 3257 | DM |
| | 3172 | (BH) = DISPLAY PAGE (VALID FOR ALPHA MODES ONLY) | F063 74F2 | 3258 | DM |
| | 3173 | ON EXIT: | 0020 | 3259 | DM |
| | 3174 | (AL) = CHAR READ | | 3260 | MIL |
| | 3175 | (AH) = ATTRIBUTE OF CHARACTER READ (ALPHA MODES ONLY) | | 3261 | PROC |
| | 3176 | (BH) = ATTRIBUTE/CHARACTER AT CURRENT CURSOR POSITION | F0A5 F0 | 3262 | STZ |
| | 3177 | (CX) = COUNT OF CHARACTERS TO WRITE | F0A6 FC | 3263 | CLD |
| | 3178 | (AL) = CHAR TO WRITE | F0A7 0A | 3264 | PUSH |
| | 3179 | (BL) = ATTRIBUTE OF CHARACTER (ALPHA/COLOR OF CHAR (GRAPHICS)) | F0A8 1E | 3265 | PUSH |
| | 3180 | SEE NOTE ON WRITE DOT FOR BIT 7 OF BL = 1. | F0A9 52 | 3266 | PUSH |
| | 3181 | (BH) = 10 WRITE CHARACTER ONLY AT CURRENT CURSOR POSITION | F0A8 51 | 3267 | PUSH |
| | 3182 | (BH) = DISPLAY PAGE (VALID FOR ALPHA MODES ONLY) | F0A8 53 | 3268 | PUSH |
| | 3183 | (CX) = COUNT OF CHARACTERS TO WRITE | F0A8 56 | 3269 | PUSH |
| | 3184 | (AL) = CHAR TO WRITE | F0A8 57 | 3270 | PUSH |
| | 3185 | FOR READ/WRITE CHARACTER INTERFACE WHILE IN GRAPHICS MODE, THE | F0A8 50 | 3271 | PUSH |
| | 3186 | CHARACTERS ARE FORMED FROM A CHARACTER GENERATOR IMAGE | F0A8 B4C4 | 3272 | PUSH |
| | 3187 | MAINTAINED IN THE SYSTEM ROM. ONLY THE 1ST 128 CHARS | F071 32E4 | 3273 | XOR |
| | 3188 | ARE CONTAINED THERE. TO READ/WRITE THE SECOND 128 CHARS, | F073 D1E0 | 3274 | AX,AX |
| | 3189 | THE USER MUST INITIALIZE THE POINTER AT INTERRUPT 1FH | F075 88F0 | 3275 | SAL |
| | 3190 | (LOCATION 0007CH) TO POINT TO THE 1X BYTE TABLE CONTAINING | F077 302080 | 3276 | MOV |
| | 3191 | THE CODE POINTS FOR THE SECOND 128 CHARS (118H-255). | F07A 72E4 | 3277 | CHP |
| | 3192 | FOR WRITE CHARACTER INTERFACE IN GRAPHICS MODE, THE REPLICATION FACTOR | F07C 58 | 3278 | JB |
| | 3193 | CONTAINED IN (CX) ON ENTRY WILL PRODUCE VALID RESULTS ONLY | F07E 54701 | 3279 | POP |
| | 3194 | FOR CHARACTERS CONTAINED ON THE SAME ROW. CONTINUATION TO | F080 8A0000 | 3280 | JMP |
| | 3195 | SUCCESSING LINES WILL NOT PRODUCE CORRECTLY. | F083 4E08 | 3281 | MOV |
| | 3196 | GRAPHICS INTERFACE | F085 800808 | 3282 | MOV |
| | 3197 | (AH) = 11 SET COLOR PALETTE | F0A8 0A0E1000 | 3283 | MOV |
| | 3198 | (BH) = PALETTE COLOR ID BEING SET (0-127) | F0A8 01F30000 | 3284 | AND |
| | 3199 | (BL) = COLOR VALUE TO BE USED WITH THAT COLOR ID | F0A9 03F3D0 | 3285 | CHP |
| | 3200 | NOTE: FOR THE CURRENT COLOR CARD, THIS ENTRY POINT HAS | F073 75D3 | 3286 | JNE |
| | 3201 | MEANING ONLY FOR 320x200 GRAPHICS. | F075 88D0 | 3287 | MOV |
| | 3202 | COLOR ID = 0 SELECTS THE BACKGROUND COLOR (0-15) | F0A8 BEC0 | 3288 | POP |
| | 3203 | COLOR ID = 1 SELECTS THE PALETTE TO BE USED: | F0A8 8A | 3289 | MOV |
| | 3204 | 0 = GREEN(1)/RED(2)/YELLOW(3) | F0A8 84544000 | 3290 | MOV |
| | 3205 | 1 = CYAN(1)/MAGENTA(2)/WHITE(3) | F0A8 2E7FAA45F0 | 3291 | JMP |
| | 3206 | IN 40X25 OR 80X25 ALPHA MODES, THE VALUE SET FOR | | 3292 | VIDEO_ID |
| | 3207 | PALETTE COLOR 0 INDICATES THE BORDER COLOR | | 3293 | ENOP |
| | 3208 | TO BE USED (VALUES 0-31, WHERE 16-31 SELECT THE | | 3294 | SET_MODE |
| | 3209 | HIGH INTENSITY BACKGROUND SET. | | 3295 | THIS ROUTINE INITIALIZES THE ATTACHMENT TO |
| | 3210 | (AH) = 12 WRITE DOT | | 3296 | THE SELECTED MODE. THE SCREEN IS BLANKED. |
| | 3211 | (CX) = ROW NUMBER | | 3297 | INRUT |
| | 3212 | (CX) = COLUMN NUMBER | | 3298 | OUTPUT |
| | 3213 | (AL) = COLOR VALUE | | 3299 | (AL) = MODE SELECTED (RANGE 0-9) |
| | 3214 | IF BIT 7 OF AL = 1, THEN THE COLOR VALUE IS EXCLUSIVE | | 3300 | NONE |
| | 3215 | OR 0 WITH THE CURRENT CONTENTS OF THE DOT | | 3301 | ----- |
| | 3216 | (AH) = 13 READ DOT | | 3302 | ----- |
| | 3217 | (DX) = ROW NUMBER | | 3303 | ----- |
| | 3218 | (CX) = COLUMN NUMBER | | 3304 | ----- |
| | 3219 | (AL) RETURNS THE DOT READ | | 3305 | ----- |
| | 3220 | ASCII TELETYPE ROUTINE FOR OUTPUT | | 3306 | ----- |
| | 3221 | (AH) = 14 WRITE TELETYPE | | 3307 | ----- |
| | 3222 | (AL) = CHAR TO WRITE | | | |
| | 3223 | (BL) = FOREGROUND COLOR IN GRAPHICS MODE | | | |
| | 3224 | (BH) = DISPLAY PAGE IN ALPHA MODE | | | |
| | 3225 | NOTE -- SCREEN WIDTH IS CONTROLLED BY PREVIOUS MODE SET | | | |
| | 3226 | | | | |
| | 3227 | | | | |
| | 3228 | | | | |
| | 3229 | | | | |
| | 3230 | | | | |
| | 3231 | | | | |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|----------------------|------|--------|----------------|------|----------------|
| F0A4 36A200A1F0619 | 3308 | | F146 EE | 3384 | |
| F0A8 1C02070607 | 3309 | | F147 42 | 3385 | OUT DX:AL |
| F0B0 00000000 | 3310 | | INC AX | 3386 | INC AX |
| 0010 | 3311 | | MOV AL:BX1 | 3387 | MOV AL:BX1 |
| | 3312 | | MOV DX:AL | 3388 | MOV DX:AL |
| F0B6 71050401F0619 | 3313 | | INC DX | 3389 | INC DX |
| F0B8 1C02070607 | 3314 | | DEC DX | 3390 | DEC DX |
| F0C0 00000000 | 3315 | | LOOP H10 | 3391 | LOOP H10 |
| | 3316 | | POP AX | 3392 | POP AX |
| F0C4 36A200A7F0664 | 3317 | | POP DS | 3393 | POP DS |
| F0C8 7020100607 | 3318 | | ASSUME DS:DATA | 3394 | ASSUME DS:DATA |
| F0D0 00000000 | 3319 | | 3395 | 3395 | |
| | 3320 | | 3396 | 3396 | |
| F0D4 6105020F10619 | 3321 | | 3397 | 3397 | |
| F0D8 190200B0C | 3322 | | | | |
| F0E0 00000000 | 3323 | | | | |
| | 3324 | | | | |
| F0E4 0000 | 3325 | | | | |
| F0E6 0010 | 3326 | | | | |
| F0E8 0040 | 3327 | | | | |
| F0EA 0040 | 3328 | | | | |
| | 3329 | | | | |
| | 3330 | | | | |
| | 3331 | | | | |
| F0EC 0000 | 3332 | | | | |
| F0EC 2605050206050 | 3333 | | | | |
| | 3334 | | | | |
| | 3335 | | | | |
| F0F4 2C080202A2E1E29 | 3336 | | | | |
| | 3337 | | | | |
| | 3338 | | | | |
| | 3339 | | | | |
| F0FC 8A0403 | 3340 | | | | |
| F0FF 0300 | 3341 | | | | |
| F101 83FF30 | 3342 | | | | |
| F104 7507 | 3343 | | | | |
| F106 B007 | 3344 | | | | |
| F108 B40403 | 3345 | | | | |
| F10B FEC3 | 3346 | | | | |
| F10D 6A00 | 3347 | | | | |
| F10F A24900 | 3348 | | | | |
| F112 0910A300 | 3349 | | | | |
| F116 1E | 3350 | | | | |
| F117 50 | 3351 | | | | |
| F118 52 | 3352 | | | | |
| F119 83C204 | 3353 | | | | |
| F11C 0A03 | 3354 | | | | |
| F11E 1E | 3355 | | | | |
| F11F 5A | 3356 | | | | |
| F120 28C0 | 3357 | | | | |
| F122 8E00 | 3358 | | | | |
| | 3359 | | | | |
| F124 C51E7400 | 3360 | | | | |
| F128 50 | 3361 | | | | |
| F129 891000 | 3362 | | | | |
| F12C 00FC02 | 3363 | | | | |
| F12F 7210 | 3364 | | | | |
| F131 0309 | 3365 | | | | |
| F133 00FC04 | 3366 | | | | |
| F136 7209 | 3367 | | | | |
| F138 0309 | 3368 | | | | |
| F13A 00FC07 | 3369 | | | | |
| F13D 7202 | 3370 | | | | |
| F13F 0309 | 3371 | | | | |
| | 3372 | | | | |
| | 3373 | | | | |
| | 3374 | | | | |
| F141 | 3375 | | | | |
| F141 50 | 3376 | | | | |
| F142 32E4 | 3377 | | | | |
| | 3378 | | | | |
| | 3379 | | | | |
| | 3380 | | | | |
| | 3381 | | | | |
| F144 | 3382 | | | | |
| F144 0A04 | 3383 | | | | |
| | 3384 | | | | |
| | 3385 | | | | |
| | 3386 | | | | |
| | 3387 | | | | |
| | 3388 | | | | |
| | 3389 | | | | |
| | 3390 | | | | |
| | 3391 | | | | |
| | 3392 | | | | |
| | 3393 | | | | |
| | 3394 | | | | |
| | 3395 | | | | |
| | 3396 | | | | |
| | 3397 | | | | |
| | 3398 | | | | |
| | 3399 | | | | |
| | 3400 | | | | |
| | 3401 | | | | |
| | 3402 | | | | |
| | 3403 | | | | |
| | 3404 | | | | |
| | 3405 | | | | |
| | 3406 | | | | |
| | 3407 | | | | |
| | 3408 | | | | |
| | 3409 | | | | |
| | 3410 | | | | |
| | 3411 | | | | |
| | 3412 | | | | |
| | 3413 | | | | |
| | 3414 | | | | |
| | 3415 | | | | |
| | 3416 | | | | |
| | 3417 | | | | |
| | 3418 | | | | |
| | 3419 | | | | |
| | 3420 | | | | |
| | 3421 | | | | |
| | 3422 | | | | |
| | 3423 | | | | |
| | 3424 | | | | |
| | 3425 | | | | |
| | 3426 | | | | |
| | 3427 | | | | |
| | 3428 | | | | |
| | 3429 | | | | |
| | 3430 | | | | |
| | 3431 | | | | |
| | 3432 | | | | |
| | 3433 | | | | |
| | 3434 | | | | |
| | 3435 | | | | |
| | 3436 | | | | |
| | 3437 | | | | |
| | 3438 | | | | |
| | 3439 | | | | |
| | 3440 | | | | |
| | 3441 | | | | |
| | 3442 | | | | |
| | 3443 | | | | |
| | 3444 | | | | |
| | 3445 | | | | |
| | 3446 | | | | |
| | 3447 | | | | |
| | 3448 | | | | |
| | 3449 | | | | |
| | 3450 | | | | |
| | 3451 | | | | |
| | 3452 | | | | |
| | 3453 | | | | |
| | 3454 | | | | |
| | 3455 | | | | |
| | 3456 | | | | |
| | 3457 | | | | |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|---------|------|---|-------------|------|---|
| F1C7 SF | 3458 | VIDEO_RETURN: | F218 EBCIFF | 3533 | CALL M16 ; OUTPUT THE VALUE TO THE 6845 |
| F1C8 SE | 3459 | POP SI | F219 C3 | 3534 | RET |
| F1C9 SB | 3460 | POP SI | | 3535 | ENDP |
| F1CA | 3461 | POP DX | | 3536 | M18 |
| F1CA 59 | 3462 | M15: POP CX | | 3537 | READ_CURSOR |
| F1CB 5A | 3463 | POP DX | | 3538 | THIS ROUTINE READS THE CURRENT CURSOR VALUE FROM THE |
| F1CC 5B | 3464 | POP DS | | 3539 | 6845, FORMATS IT, AND SENDS IT BACK TO THE CALLER |
| F1CD 07 | 3465 | POP ES | | 3540 | INPUT |
| F1CE CF | 3466 | POP ES | | 3541 | BM - PAGE OF CURSOR |
| | 3467 | IRET | | 3542 | OUTPUT |
| | 3468 | SET_MODE ENDP | | 3543 | DX - ROW, COLUMN OF THE CURRENT CURSOR POSITION |
| | 3469 | | | 3544 | CX - CURRENT CURSOR MODE |
| | 3470 | SET_CTYPE | | 3545 | READ_CURSOR PROC NEAR |
| | 3471 | THIS ROUTINE SETS THE CURSOR VALUE | | 3546 | MOV BL,BH |
| | 3472 | INPUT (CX) HAS CURSOR VALUE CH-START LINE, CL-STOP LINE | | 3547 | XOR BH,BH |
| | 3473 | OUTPUT | | 3548 | SAL BX,1 ; WORD OFFSET |
| | 3474 | NONE | | 3549 | MOV DX,16X-OFFSET CURSOR_POSH1 |
| | 3475 | | | 3550 | MOV CX,CURSOR_MODE |
| | 3476 | | | 3551 | POP DI |
| | 3477 | SET_CTYPE PROC NEAR | | 3552 | POP SI |
| | 3478 | MOV AH,10 ; 6845 REGISTER FOR CURSOR SET | | 3553 | POP DX |
| | 3479 | MOV CURSOR_MODE,CX ; SAVE IN DATA AREA | | 3554 | POP AX |
| | 3480 | CALL M16 ; OUTPUT CX REG | | 3555 | POP AX |
| | 3481 | JMP VIDEO_RETURN | | 3556 | POP DS |
| | 3482 | | | 3557 | POP ES |
| | 3483 | THIS ROUTINE OUTPUTS THE CX REGISTER TO THE 6845 REGS NAMED IN AH | | 3558 | IRET |
| | 3484 | | | 3559 | READ_CURSOR ENDP |
| | 3485 | M16: | | 3560 | ACT_DISP_PAGE |
| | 3486 | MOV DX,ADDR_6845 ; ADDRESS REGISTER | | 3561 | MOV CX,CRT_LEN ; GET SAVED LENGTH OF REGEN BUFFER |
| | 3487 | MOV AL,AH ; GET VALUE | | 3562 | CONV AL TO WORD |
| | 3488 | OUT DX,AL ; REGISTER SET | | 3563 | SAVE PAGE VALUE |
| | 3489 | INC DX ; DATA REGISTER | | 3564 | DISPLAY PAGE TIMES REGEN LENGTH |
| | 3490 | MOV AL,CH ; DATA | | 3565 | START ADDRESS FOR LATER REQUIREMENTS |
| | 3491 | OUT DX,AL ; DATA | | 3566 | START ADDRESS TO CX |
| | 3492 | DEC DX | | 3567 | DIVIDE BY 2 FOR 6845 HANDLING |
| | 3493 | MOV AL,AH ; POINT TO OTHER DATA REGISTER | | 3568 | 6845 REGISTER FOR START ADDRESS |
| | 3494 | INC AL ; SET FOR SECOND REGISTER | | 3569 | CALL M16 |
| | 3495 | INC DX | | 3570 | POP BX |
| | 3496 | MOV AL,CL ; SECOND DATA VALUE | | 3571 | SAL DX,1 ; RECOVER PAGE VALUE |
| | 3497 | INC DX | | 3572 | MOV AX,16X + OFFSET CURSOR_POSH1 ; SET CURSOR FOR THIS PAGE |
| | 3498 | OUT DX,AL ; ALL DONE | | 3573 | CALL M18 ; SET THE CURSOR POSITION |
| | 3499 | RET | | 3574 | JMP VIDEO_RETURN |
| | 3500 | SET_CTYPE ENDP | | 3575 | ACT_DISP_PAGE ENDP |
| | 3501 | | | 3576 | SET COLOR |
| | 3502 | SET_CPOS | | 3577 | THIS ROUTINE WILL ESTABLISH THE BACKGROUND COLOR, THE OVERSCAN COLOR, |
| | 3503 | THIS ROUTINE SETS THE CURRENT CURSOR POSITION TO THE | | 3578 | AND THE FOREGROUND COLOR SET FOR MEDIUM RESOLUTION GRAPHICS |
| | 3504 | NEW X-Y VALUES PASSED | | 3579 | INPUT |
| | 3505 | INPUT | | 3580 | BM HAS COLOR ID |
| | 3506 | DX - ROW, COLUMN OF NEW CURSOR | | 3581 | IF BM=0, THE BACKGROUND COLOR VALUE IS SET |
| | 3507 | BM - DISPLAY PAGE OF CURSOR | | 3582 | FROM THE LOW BITS OF BL (0-31) |
| | 3508 | OUTPUT | | 3583 | IF BM=1, THE PALLETTE SELECTION IS MADE |
| | 3509 | CURSOR IS SET AT 6845 IF DISPLAY PAGE IS CURRENT DISPLAY | | 3584 | BASED ON THE LOW BIT OF BL: |
| | 3510 | | | 3585 | 0 = GREEN, RED, YELLOW FOR COLORS 1,2,3 |
| | 3511 | SET_CPOS PROC NEAR | | 3586 | 1 = BLUE, CYAN, MAGENTA FOR COLORS 1,2,3 |
| | 3512 | MOV CL,BH | | 3587 | 1BL HAS THE COLOR VALUE TO BE USED |
| | 3513 | XOR CH,CH | | 3588 | THE COLOR SELECTION IS UPDATED |
| | 3514 | SAL CX,1 | | 3589 | SET_COLOR PROC NEAR |
| | 3515 | MOV SI,CX ; USE INDEX REGISTER | | 3590 | MOV DX,ADDR_6845 ; I/O PORT FOR PALLETTE |
| | 3516 | MOV SI,OFFSET CURSOR_POSH1,DX ; SAVE THE POINTER | | 3591 | ADD DX,5 ; OVERSCAN PORT |
| | 3517 | MOV SI,ACTIVE_PAGE,DX | | 3592 | MOV AL,CRT_PALLETTE ; GET THE CURRENT PALLETTE VALUE |
| | 3518 | JNZ M17 | | 3593 | OR BH,BH ; IS THIS COLOR 0? |
| | 3519 | MOV AX,DX ; SET_CPOS_RETURN | | | |
| | 3520 | MOV AX,DX ; GET ROW/COLUMN TO AX | | | |
| | 3521 | CALL M18 ; CURSOR SET | | | |
| | 3522 | JMP VIDEO_RETURN | | | |
| | 3523 | SET_CPOS ENDP | | | |
| | 3524 | SET CURSOR POSITION, AX HAS ROW/COLUMN FOR CURSOR | | | |
| | 3525 | | | | |
| | 3526 | | | | |
| | 3527 | M18 PROC NEAR | | | |
| | 3528 | CALL POSITION | | | |
| | 3529 | MOV CX,AX ; DETERMINE LOCATION IN REGEN BUFFER | | | |
| | 3530 | ADD CX,CRT_START | | | |
| | 3531 | SAR CX,1 ; ADD IN THE START ADDRESS FOR THIS PAGE | | | |
| | 3532 | MOV AH,14 ; DIVIDE BY 2 FOR CHAR ONLY COUNT | | | |
| | | REGISTER NUMBER FOR CURSOR | | | |

| LOC OBJ | LINE | SOURCE | JNZ | MC0 | LOC OBJ | LINE | SOURCE |
|---------------|------|--------|-----|-----|----------------|------|--------|
| F260 759E | 3608 | | | | F2A1 720B | 3664 | |
| | 3609 | | | | F2A3 80FC57 | 3665 | |
| | 3610 | | | | F2A6 7403 | 3666 | |
| | 3611 | | | | F2A8 E9F301 | 3667 | |
| | 3612 | | | | F2AB | 3668 | |
| | 3613 | | | | F2AB 53 | 3669 | |
| | 3614 | | | | F2AC 80C1 | 3670 | |
| | 3615 | | | | F2AE 82390B | 3691 | |
| | 3616 | | | | F2B1 7433 | 3692 | |
| | 3617 | | | | F2B3 01F0 | 3693 | |
| | 3618 | | | | F2B5 8466 | 3694 | |
| | 3619 | | | | F2B7 2A63 | 3695 | |
| | 3620 | | | | F2B9 | 3696 | |
| | 3621 | | | | F2B9 E97500 | 3697 | |
| | 3622 | | | | F2BC 03F5 | 3698 | |
| | 3623 | | | | F2BE 03F5 | 3699 | |
| | 3624 | | | | F2C0 FECC | 3700 | |
| | 3625 | | | | F2C2 75F5 | 3701 | |
| | 3626 | | | | F2C4 | 3702 | |
| | 3627 | | | | F2C4 56 | 3703 | |
| | 3628 | | | | F2C5 8620 | 3704 | |
| | 3629 | | | | F2C7 | 3705 | |
| | 3630 | | | | F2C7 E87000 | 3706 | |
| | 3631 | | | | F2CA 03F0 | 3707 | |
| | 3632 | | | | F2CC FE0B | 3708 | |
| | 3633 | | | | F2CE 78F7 | 3709 | |
| | 3634 | | | | F2D0 | 3710 | |
| | 3635 | | | | F2D0 8A6A00 | 3711 | |
| | 3636 | | | | F2D3 80C0 | 3712 | |
| | 3637 | | | | F2D5 803E49007 | 3713 | |
| | 3638 | | | | F2DA 7407 | 3714 | |
| | 3639 | | | | F2DC A68500 | 3715 | |
| | 3640 | | | | F2DE B0A003 | 3716 | |
| | 3641 | | | | F2EE EE | 3717 | |
| | 3642 | | | | F2E3 | 3718 | |
| | 3643 | | | | F2E3 E9E1FE | 3719 | |
| | 3644 | | | | F2E6 | 3720 | |
| | 3645 | | | | F2E6 84DE | 3721 | |
| | 3646 | | | | F2E8 E8DA | 3722 | |
| | 3647 | | | | | 3723 | |
| | 3648 | | | | | 3724 | |
| | 3649 | | | | | 3725 | |
| | 3650 | | | | | 3726 | |
| | 3651 | | | | | 3727 | |
| | 3652 | | | | | 3728 | |
| | 3653 | | | | | 3729 | |
| | 3654 | | | | | 3730 | |
| | 3655 | | | | | 3731 | |
| | 3656 | | | | | 3732 | |
| | 3657 | | | | | 3733 | |
| | 3658 | | | | | 3734 | |
| | 3659 | | | | | 3735 | |
| | 3660 | | | | | 3736 | |
| | 3661 | | | | | 3737 | |
| | 3662 | | | | | 3738 | |
| | 3663 | | | | | 3739 | |
| | 3664 | | | | | 3740 | |
| | 3665 | | | | | 3741 | |
| | 3666 | | | | | 3742 | |
| | 3667 | | | | | 3743 | |
| | 3668 | | | | | 3744 | |
| | 3669 | | | | | 3745 | |
| | 3670 | | | | | 3746 | |
| | 3671 | | | | | 3747 | |
| | 3672 | | | | | 3748 | |
| | 3673 | | | | | 3749 | |
| | 3674 | | | | | 3750 | |
| | 3675 | | | | | 3751 | |
| | 3676 | | | | | 3752 | |
| | 3677 | | | | | 3753 | |
| | 3678 | | | | | 3754 | |
| | 3679 | | | | | 3755 | |
| | 3680 | | | | | 3756 | |
| | 3681 | | | | | 3757 | |
| | 3682 | | | | | 3758 | |
| | 3683 | | | | | 3759 | |
| F270 | 3684 | | | | F2E6 52 | 3735 | |
| F270 240F | 3685 | | | | F2F9 B0A003 | 3736 | |
| F272 00E8 | 3686 | | | | F2FC 50 | 3737 | |
| F274 73F3 | 3687 | | | | F2FD | 3738 | |
| F276 0C20 | 3688 | | | | F2FD EC | 3739 | |
| F276 E8EF | 3689 | | | | F2FE A008 | 3740 | |
| | 3690 | | | | F300 74F8 | 3741 | |
| | 3691 | | | | F302 80E5 | 3742 | |
| | 3692 | | | | F304 B0A003 | 3743 | |
| | 3693 | | | | F307 EE | 3744 | |
| | 3694 | | | | F308 58 | 3745 | |
| | 3695 | | | | F309 5A | 3746 | |
| | 3696 | | | | F30A E97E7F | 3747 | |
| | 3697 | | | | F30B 030A4E0B | 3748 | |
| | 3698 | | | | F311 80F8 | 3749 | |
| | 3699 | | | | F313 80F0 | 3750 | |
| | 3700 | | | | F315 2B01 | 3751 | |
| | 3701 | | | | F317 FE0A | 3752 | |
| | 3702 | | | | F319 FE02 | 3753 | |
| | 3703 | | | | F31B 3EED | 3754 | |
| | 3704 | | | | F31D 80E4A00 | 3755 | |
| | 3705 | | | | F321 03ED | 3756 | |
| | 3706 | | | | F323 8AC3 | 3757 | |
| | 3707 | | | | F325 F62A4A00 | 3758 | |
| | 3708 | | | | F329 03C0 | 3759 | |
| F27A 8A26A00 | 3684 | | | | F2E6 52 | 3735 | |
| F27A 40A900 | 3685 | | | | F2F9 B0A003 | 3736 | |
| F27B 8A3E6100 | 3686 | | | | F2FC 50 | 3737 | |
| F280 5F | 3687 | | | | F2FD | 3738 | |
| F280 8E | 3688 | | | | F2FD EC | 3739 | |
| F287 5F | 3689 | | | | F2FE A008 | 3740 | |
| F288 E93FF7 | 3690 | | | | F300 74F8 | 3741 | |
| | 3691 | | | | F302 80E5 | 3742 | |
| | 3692 | | | | F304 B0A003 | 3743 | |
| | 3693 | | | | F307 EE | 3744 | |
| | 3694 | | | | F308 58 | 3745 | |
| | 3695 | | | | F309 5A | 3746 | |
| | 3696 | | | | F30A E97E7F | 3747 | |
| | 3697 | | | | F30B 030A4E0B | 3748 | |
| | 3698 | | | | F311 80F8 | 3749 | |
| | 3699 | | | | F313 80F0 | 3750 | |
| | 3700 | | | | F315 2B01 | 3751 | |
| | 3701 | | | | F317 FE0A | 3752 | |
| | 3702 | | | | F319 FE02 | 3753 | |
| | 3703 | | | | F31B 3EED | 3754 | |
| | 3704 | | | | F31D 80E4A00 | 3755 | |
| | 3705 | | | | F321 03ED | 3756 | |
| | 3706 | | | | F323 8AC3 | 3757 | |
| | 3707 | | | | F325 F62A4A00 | 3758 | |
| | 3708 | | | | F329 03C0 | 3759 | |
| F27C | 3684 | | | | F2E6 52 | 3735 | |
| F27C 8A08 | 3685 | | | | F2F9 B0A003 | 3736 | |
| F27E 80FC04 | 3686 | | | | F2FC 50 | 3737 | |
| | 3687 | | | | F2FD | 3738 | |
| | 3688 | | | | F2FD EC | 3739 | |
| | 3689 | | | | F2FE A008 | 3740 | |
| | 3690 | | | | F300 74F8 | 3741 | |
| | 3691 | | | | F302 80E5 | 3742 | |
| | 3692 | | | | F304 B0A003 | 3743 | |
| | 3693 | | | | F307 EE | 3744 | |
| | 3694 | | | | F308 58 | 3745 | |
| | 3695 | | | | F309 5A | 3746 | |
| | 3696 | | | | F30A E97E7F | 3747 | |
| | 3697 | | | | F30B 030A4E0B | 3748 | |
| | 3698 | | | | F311 80F8 | 3749 | |
| | 3699 | | | | F313 80F0 | 3750 | |
| | 3700 | | | | F315 2B01 | 3751 | |
| | 3701 | | | | F317 FE0A | 3752 | |
| | 3702 | | | | F319 FE02 | 3753 | |
| | 3703 | | | | F31B 3EED | 3754 | |
| | 3704 | | | | F31D 80E4A00 | 3755 | |
| | 3705 | | | | F321 03ED | 3756 | |
| | 3706 | | | | F323 8AC3 | 3757 | |
| | 3707 | | | | F325 F62A4A00 | 3758 | |
| | 3708 | | | | F329 03C0 | 3759 | |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|-------------|------|---|--|------|---|
| F328 06 | 3760 | PUSH ES | F376 E8D | 3834 | JMP N14 |
| F32C 1F | 3761 | POP DS | SCROLL_DOWN | 3835 | ENDP |
| F32D 00F800 | 3762 | CHP BL,0 | 1: READ_AC_CURRENT | 3836 | 1:----- |
| F330 C3 | 3763 | RET | 1: THIS ROUTINE READS THE ATTRIBUTE AND CHARACTER AT THE CURRENT | 3837 | 1: CURSOR POSITION AND RETURNS THEM TO THE CALLER |
| | 3764 | SCROLL_POSITION ENDP | 1: INPUT | 3838 | 1:----- |
| F331 | 3765 | 1:----- MOVE_ROM | 3839 | 3840 | 1: (AH) = CURRENT CRT MODE |
| F331 BACA | 3766 | MOV CL,DL | 3841 | 3842 | 1: (BH) = DISPLAY PAGE (ALPHA MODES ONLY) |
| F333 56 | 3767 | PUSH SI | 3843 | 3844 | 1: (DS) = DATA SEGMENT |
| F334 57 | 3768 | PUSH DI | 3845 | 3846 | 1: (ES) = REGEN SEGMENT |
| F335 F3 | 3769 | REP MOVSW | 3847 | 3848 | 1: (AL) = CHAR READ |
| F336 A5 | 3770 | REP MOVSW | 3849 | 3850 | 1: (AH) = ATTRIBUTE READ |
| F337 5F | 3771 | POP DI | 3851 | 3852 | 1: ASSUME CS:CODE,DS:DATA,ES:DATA |
| F338 5E | 3772 | POP SI | 3853 | 3854 | 1: READ_AC_CURRENT PROC NEAR |
| F339 C3 | 3773 | POP SI | 3855 | 3856 | 1: CHP AH,4 |
| | 3774 | RET | 3857 | 3858 | 1: JC P1 |
| | 3775 | 1: RECOVER ADDRESSES | 3859 | 3860 | 1: CHP AH,7 |
| | 3776 | 1: N10 ENDP | 3861 | 3862 | 1: JE P1 |
| F33A | 3777 | 1:----- CLEAR_ROM | 3863 | 3864 | 1: JNE P1 |
| F33A BACA | 3778 | MOV CL,DL | 3865 | 3866 | 1: JNE P1 |
| F33C 57 | 3779 | MOV CL,DL | 3867 | 3868 | 1: JNE P1 |
| F33D 57 | 3780 | PUSH DI | 3869 | 3870 | 1: JNE P1 |
| F33D F3 | 3781 | REP STOSW | 3871 | 3872 | 1: JNE P1 |
| F33E A0 | 3782 | POP DI | 3873 | 3874 | 1: JNE P1 |
| F33F 5F | 3783 | POP DI | 3875 | 3876 | 1: JNE P1 |
| F340 C3 | 3784 | RET | 3877 | 3878 | 1: JNE P1 |
| | 3785 | 1: N11 ENDP | 3879 | 3880 | 1: JNE P1 |
| F33A | 3786 | 1:----- SCROLL_DOWN | 3881 | 3882 | 1: JNE P1 |
| F33A BACA | 3787 | 1: THIS ROUTINE MOVES THE CHARACTERS WITHIN A DEFINED | 3883 | 3884 | 1: JNE P1 |
| F33C 57 | 3788 | 1: BLOCK DOWN ON THE SCREEN, FILLING THE TOP LINES | 3885 | 3886 | 1: JNE P1 |
| F33D 57 | 3789 | 1: WITH A DEFINED CHARACTER | 3887 | 3888 | 1: JNE P1 |
| F33D F3 | 3790 | 1: INPUT | 3889 | 3890 | 1: JNE P1 |
| F33E A0 | 3791 | 1: (AH) = CURRENT CRT MODE | 3891 | 3892 | 1: JNE P1 |
| F33F 5F | 3792 | 1: (AL) = NUMBER OF LINES TO SCROLL | 3893 | 3894 | 1: JNE P1 |
| F340 C3 | 3793 | 1: (CX) = UPPER LEFT CORNER OF REGION | 3895 | 3896 | 1: JNE P1 |
| | 3794 | 1: (DX) = LOWER RIGHT CORNER OF REGION | 3897 | 3898 | 1: JNE P1 |
| | 3795 | 1: (BH) = FILL CHARACTER | 3899 | 3900 | 1: JNE P1 |
| | 3796 | 1: (DS) = DATA SEGMENT | 3901 | 3902 | 1: JNE P1 |
| | 3797 | 1: (ES) = REGEN SEGMENT | 3903 | 3904 | 1: JNE P1 |
| | 3798 | 1: N12 ENDP | 3905 | 3906 | 1: JNE P1 |
| | 3799 | 1: N13 ENDP | 3907 | 3908 | 1: JNE P1 |
| | 3800 | 1: N14 ENDP | 3909 | 3910 | 1: JNE P1 |
| F341 | 3801 | 1:----- SCROLL_DOWN | 3911 | 3912 | 1: JNE P1 |
| F341 F0 | 3802 | 1: LINE COUNT TO BL | 3913 | 3914 | 1: JNE P1 |
| F342 8AD8 | 3803 | MOV BL,AL | 3915 | 3916 | 1: JNE P1 |
| F344 80FC04 | 3804 | CHP AH,4 | 3917 | 3918 | 1: JNE P1 |
| F347 7208 | 3805 | JC N12 | 3919 | 3920 | 1: JNE P1 |
| F349 60FC07 | 3806 | CHP AH,7 | 3921 | 3922 | 1: JNE P1 |
| F34C 7403 | 3807 | JE N12 | 3923 | 3924 | 1: JNE P1 |
| F34E E9A601 | 3808 | JNE N12 | 3925 | 3926 | 1: JNE P1 |
| F351 | 3809 | 1: CONTINUE DOWN | 3927 | 3928 | 1: JNE P1 |
| F351 53 | 3810 | PUSH BX | 3929 | 3930 | 1: JNE P1 |
| F352 08C2 | 3811 | MOV AX,DX | 3931 | 3932 | 1: JNE P1 |
| F354 E807FF | 3812 | CALL SCROLL_POSITION | 3933 | 3934 | 1: JNE P1 |
| F357 7420 | 3813 | JZ N14 | 3935 | 3936 | 1: JNE P1 |
| F359 28F0 | 3814 | SUB SI,AX | 3937 | 3938 | 1: JNE P1 |
| F35B 84C6 | 3815 | MOV AH,0H | 3939 | 3940 | 1: JNE P1 |
| F35D 2A83 | 3816 | SUB AH,BL | 3941 | 3942 | 1: JNE P1 |
| F35F | 3817 | 1: N13 ENDP | 3943 | 3944 | 1: JNE P1 |
| F35F E8C7FF | 3818 | CALL N10 | 3945 | 3946 | 1: JNE P1 |
| F362 28F5 | 3819 | SUB SI,BP | 3947 | 3948 | 1: JNE P1 |
| F364 28F0 | 3820 | SUB DI,BP | 3949 | 3950 | 1: JNE P1 |
| F366 FECC | 3821 | DEC AH | 3951 | 3952 | 1: JNE P1 |
| F368 75F5 | 3822 | JNZ N13 | 3953 | 3954 | 1: JNE P1 |
| F36A | 3823 | 1: N14 ENDP | 3955 | 3956 | 1: JNE P1 |
| F36A 58 | 3824 | POP AX | 3957 | 3958 | 1: JNE P1 |
| F36B B020 | 3825 | MOV AL, | 3959 | 3960 | 1: JNE P1 |
| F36D | 3826 | 1: N15 ENDP | 3961 | 3962 | 1: JNE P1 |
| F36D E8C7FF | 3827 | CALL N11 | 3963 | 3964 | 1: JNE P1 |
| F370 28F0 | 3828 | SUB DI,BP | 3965 | 3966 | 1: JNE P1 |
| F372 FECC | 3829 | DEC BL | 3967 | 3968 | 1: JNE P1 |
| F374 75F7 | 3830 | JNZ N15 | 3969 | 3970 | 1: JNE P1 |
| F376 E8C7FF | 3831 | JMP N5 | 3971 | 3972 | 1: JNE P1 |
| F378 | 3832 | 1: N16 ENDP | 3973 | 3974 | 1: JNE P1 |
| F378 8AD8 | 3833 | MOV BL,0H | 3975 | 3976 | 1: JNE P1 |

| LOC OBJ | LOC SOURCE | LINE | LOC OBJ | LINE | SOURCE |
|---------------|------------|------|-------------|------|--------|
| F3C3 | | 3910 | F41C 74FB | 3986 | |
| F3C3 00FC04 | | 3911 | F41E 8AC3 | 3987 | |
| F3C6 7E08 | | 3912 | F420 AA | 3988 | |
| F3C8 00FC07 | | 3913 | F421 47 | 3989 | |
| F3C8 7403 | | 3914 | F422 E2E8 | 3990 | |
| F3C0 E90101 | | 3915 | F424 E9A0FD | 3991 | |
| F3D0 | | 3916 | | 3992 | |
| F3D0 84E3 | | 3917 | | 3993 | |
| F3D2 50 | | 3918 | | 3994 | |
| F3D3 51 | | 3919 | | 3995 | |
| F3D6 E000FF | | 3920 | | 3996 | |
| F3D7 80FB | | 3921 | | 3997 | |
| F3D9 55 | | 3922 | | 3998 | |
| F3DA 58 | | 3923 | | 3999 | |
| F3DB | | 3924 | | 4000 | |
| | | 3925 | | 4001 | |
| | | 3926 | | 4002 | |
| | | 3927 | | 4003 | |
| | | 3928 | | 4004 | |
| F3D8 0B166300 | | 3929 | | 4005 | |
| F3D9 83C206 | | 3930 | | 4006 | |
| F3E2 | | 3931 | | 4007 | |
| F3E2 EC | | 3932 | | 4008 | |
| F3E3 A001 | | 3933 | | 4009 | |
| F3E5 75FB | | 3934 | | 4010 | |
| F3E7 FA | | 3935 | | 4011 | |
| F3E8 | | 3936 | | 4012 | |
| F3E8 EC | | 3937 | | 4013 | |
| F3E9 A001 | | 3938 | | 4014 | |
| F3E9 74FB | | 3939 | | 4015 | |
| F3D0 88C3 | | 3940 | | 4016 | |
| F3EF AB | | 3941 | | 4017 | |
| F3F0 FB | | 3942 | | 4018 | |
| F3F1 E2E6 | | 3943 | | 4019 | |
| F3F3 E701FD | | 3944 | | 4020 | |
| | | 3945 | | 4021 | |
| | | 3946 | | 4022 | |
| | | 3947 | | 4023 | |
| | | 3948 | | 4024 | |
| | | 3949 | | 4025 | |
| | | 3950 | | 4026 | |
| | | 3951 | | 4027 | |
| | | 3952 | | 4028 | |
| | | 3953 | | 4029 | |
| | | 3954 | | 4030 | |
| | | 3955 | | 4031 | |
| | | 3956 | | 4032 | |
| | | 3957 | | 4033 | |
| | | 3958 | | 4034 | |
| | | 3959 | | 4035 | |
| | | 3960 | | 4036 | |
| F3F6 00FC04 | | 3961 | | 4037 | |
| F3F9 7208 | | 3962 | | 4038 | |
| F3FB 00FC07 | | 3963 | | 4039 | |
| F3FE 7403 | | 3964 | | 4040 | |
| F400 E97E01 | | 3965 | | 4041 | |
| F403 | | 3966 | | 4042 | |
| F403 50 | | 3967 | | 4043 | |
| F404 51 | | 3968 | | 4044 | |
| F405 E000FF | | 3969 | | 4045 | |
| F406 00FB | | 3970 | | 4046 | |
| F40A 59 | | 3971 | | 4047 | |
| F40B 58 | | 3972 | | 4048 | |
| F40C | | 3973 | | 4049 | |
| | | 3974 | | 4050 | |
| | | 3975 | | 4051 | |
| | | 3976 | | 4052 | |
| | | 3977 | | 4053 | |
| F40C 0B166300 | | 3978 | | 4054 | |
| F410 83C206 | | 3979 | | 4055 | |
| F413 | | 3980 | | 4056 | |
| F413 EC | | 3981 | | 4057 | |
| F414 A001 | | 3982 | | 4058 | |
| F416 75FB | | 3983 | | 4059 | |
| F418 FA | | 3984 | | 4060 | |
| F419 | | 3985 | | 4061 | |
| F419 EC | | | | | |
| F41A A001 | | | | | |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|-------------|------|--------------|-----------------|------|---|
| F400 80E2FE | 4062 | AND DL,DFEH | F447 2801 | 4137 | SUB DX,CK |
| F403 F6E2 | 4063 | MUL DL | F449 81C2D101 | 4138 | ADD DX,101H |
| F405 5A | 4064 | POP DX | F44D D0E6 | 4139 | SAL DH,1 |
| F406 FAC201 | 4065 | TEST DL,1 | F44F D0E6 | 4140 | SAL DH,1 |
| F409 7403 | 4066 | JZ R4 | | 4141 | |
| F40B 550020 | 4067 | ADD AX,2000H | | 4142 | J----- DETERMINE CRT MODE |
| F40E | 4068 | | | 4143 | |
| F40E 80F0 | 4069 | MOV SI,AX | F481 803E490006 | 4144 | CHP CRT_MODE,6 |
| F470 36 | 4070 | POP AX | F486 7304 | 4145 | JNC R7 |
| F471 8801 | 4071 | MOV DX,CK | | 4146 | |
| | 4072 | | | 4147 | J----- MEDIUM RES UP |
| | 4073 | | | 4148 | SAL DL,1 |
| | 4074 | | | 4149 | SAL DL,1 |
| | 4075 | | | 4150 | J----- DETERMINE THE SOURCE ADDRESS IN THE BUFFER |
| | 4076 | | | 4151 | R7: |
| | 4077 | | | 4152 | PUSH ES |
| | 4078 | | | 4153 | POP OS |
| | 4079 | | | 4154 | SUB CH,CH |
| | 4080 | | | 4155 | SAL BL,1 |
| | 4081 | | | 4156 | SAL BL,1 |
| | 4082 | | | 4157 | SAL BL,1 |
| | 4083 | | | 4158 | JZ R11 |
| | 4084 | | | 4159 | MOV AL,BL |
| | 4085 | | | 4160 | MOV AH,BL |
| | 4086 | | | 4161 | MUL AH |
| | 4087 | | | 4162 | MOV SI,DI |
| | 4088 | | | 4163 | ADD SI,AX |
| | 4089 | | | 4164 | MOV AH,OH |
| | 4090 | | | 4165 | SUB AH,BL |
| | 4091 | | | 4166 | J----- LOOP THROUGH, MOVING ONE ROW AT A TIME, BOTH EVEN AND ODD FIELDS |
| | 4092 | | | 4167 | R8: |
| | 4093 | | | 4168 | CALL R17 |
| | 4094 | | | 4169 | SUB SI,2000H-80 |
| | 4095 | | | 4170 | SUB DI,2000H-80 |
| | 4096 | | | 4171 | DEC AH |
| | 4097 | | | 4172 | JNC R8 |
| | 4098 | | | 4173 | JNC R8 |
| | 4099 | | | 4174 | J----- FILL IN THE VACATED LINE(S) |
| | 4100 | | | 4175 | R9: |
| | 4101 | | | 4176 | MOV AL,BH |
| | 4102 | | | 4177 | CALL R18 |
| | 4103 | | | 4178 | DEC BL |
| | 4104 | | | 4179 | SUB DI,2000H-80 |
| | 4105 | | | 4180 | DEC BL |
| | 4106 | | | 4181 | JNC R10 |
| | 4107 | | | 4182 | JMP VIDEO_RETURN |
| | 4108 | | | 4183 | |
| | 4109 | | | 4184 | |
| | 4110 | | | 4185 | |
| | 4111 | | | 4186 | |
| | 4112 | | | 4187 | |
| | 4113 | | | 4188 | |
| | 4114 | | | 4189 | |
| | 4115 | | | 4190 | |
| | 4116 | | | 4191 | |
| | 4117 | | | 4192 | |
| | 4118 | | | 4193 | |
| | 4119 | | | 4194 | |
| | 4120 | | | 4195 | |
| | 4121 | | | 4196 | |
| | 4122 | | | 4197 | |
| | 4123 | | | 4198 | |
| | 4124 | | | 4199 | |
| | 4125 | | | 4200 | |
| | 4126 | | | 4201 | |
| | 4127 | | | 4202 | |
| | 4128 | | | 4203 | |
| | 4129 | | | 4204 | |
| | 4130 | | | 4205 | |
| | 4131 | | | 4206 | |
| | 4132 | | | 4207 | |
| | 4133 | | | 4208 | |
| | 4134 | | | 4209 | |
| | 4135 | | | 4210 | |
| | 4136 | | | 4211 | |
| | | | | 4212 | |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|----------------|------|---|---------------|------|---|
| 74FF 08FA | 4213 | MOV DI, AX | F568 F3 | 4287 | REP MOVSB ; MOVE THE ODD FIELD |
| | 4214 | | F56C A4 | 4288 | POP DI ; POINTERS BACK |
| | 4215 | ;----- DETERMINE SIZE OF WINDOW | F56E 5E | 4289 | RET ; RETURN TO CALLER |
| F581 2B01 | 4216 | SUB DX, CX | | 4291 | ENDP |
| F583 61C0101 | 4217 | ADD DX, 10H | | 4292 | ;----- CLEAR A SINGLE ROW |
| F587 00E6 | 4218 | SAL DX, 1 | F570 0A0A | 4293 | |
| F589 00E6 | 4219 | SAL DX, 1 | F572 57 | 4294 | RIB PROC NEAR |
| | 4220 | ; AND EVEN/ODD ROWS | F573 F3 | 4295 | MOV CL, DL ; NUMBER OF BYTES IN FIELD |
| | 4221 | | F574 AA | 4296 | PUSH DI ; SAVE POINTER |
| | 4222 | ;----- DETERMINE CRT MODE | F575 5F | 4297 | REP STOSB ; STORE THE NEW VALUE |
| F580 603E49006 | 4223 | CMP CRT_MODE, 6 | F576 81C70020 | 4299 | POP DI ; POINTER BACK |
| F510 7305 | 4224 | JNC R12 | F57A 57 | 4300 | AOD 01, 2000H ; POINT TO ODD FIELD |
| | 4225 | ; FINE_SOURCE_DOWN | 4301 | 4302 | MOV CL, DL |
| | 4226 | | 4303 | 4304 | REP STOSB ; FILL THE ODD FILLED |
| F512 00E2 | 4227 | ;----- MEDIUM RES DOWN | 4305 | 4306 | POP DI ; RETURN TO CALLER |
| F514 01E7 | 4228 | SAL DL, 1 | | 4307 | |
| F516 47 | 4229 | INC DI | F57B 0A0A | 4308 | ; GRAPHICS WRITE |
| | 4230 | | F57D F3 | 4309 | ; THIS ROUTINE WRITES THE ASCII CHARACTER TO THE CURRENT |
| | 4231 | | F57E AA | 4310 | ; POSITION ON THE SCREEN. |
| F517 | 4232 | ;----- DETERMINE THE SOURCE ADDRESS IN THE BUFFER | 4311 | 4312 | ; ENTRY -- |
| F517 06 | 4233 | R12: | 4313 | 4314 | AL = CHARACTER TO WRITE |
| F518 1F | 4234 | PUSH ES | 4315 | 4316 | BL = COLOR ATTRIBUTE TO BE USED FOR FOREGROUND COLOR |
| F519 24E0 | 4235 | POP CH, CH | 4317 | 4318 | IF BIT 7 IS SET, THE CHAR IS XOR'D INTO THE REGEN BUFFER |
| F51B 61C7F000 | 4236 | SUB CH, CH | 4319 | 4320 | 0 IS USED FOR THE BACKGROUND COLOR |
| F51F 00E3 | 4237 | SAL BL, 1 | 4321 | 4322 | CX = NUMBER OF CHARS TO WRITE |
| F521 00E3 | 4238 | SAL BL, 1 | 4323 | 4324 | DS = DATA SEGMENT |
| F523 742E | 4239 | JZ R16 | 4325 | 4326 | ES = REGEN SEGMENT |
| F525 84C3 | 4240 | MOV AL, BL | 4327 | 4328 | NOTHING IS RETURNED |
| F527 8450 | 4241 | MOV AH, 80 | 4329 | 4330 | ; GRAPHICS READ |
| F529 FAE4 | 4242 | MUL AH | 4331 | 4332 | THIS ROUTINE READS THE ASCII CHARACTER AT THE CURRENT CURSOR |
| F52B 08F7 | 4243 | MOV SI, 01 | 4333 | 4334 | POSITION ON THE SCREEN BY MATCHING THE ODS ON THE SCREEN TO THE |
| F52D 20F0 | 4244 | SUB SI, AX | 4335 | 4336 | CHARACTER GENERATOR CODE POINTS |
| F52F 04E6 | 4245 | MOV AH, 0H | 4337 | 4338 | ENTRY -- |
| F531 24E3 | 4246 | SUB AH, BL | 4339 | 4340 | NONE (0 IS ASSUMED AS THE BACKGROUND COLOR |
| | 4247 | | 4341 | 4342 | AL = CHARACTER READ AT THAT POSITION (0 RETURNED IF NONE FOUND) |
| | 4248 | ;----- LOOP THROUGH, MOVING ONE ROW AT A TIME, BOTH EVEN AND ODD FIELDS | 4343 | 4344 | FOR BOTH ROUTINES, THE IMAGES USED TO FORM CHARS ARE CONTAINED IN ROM |
| F533 | 4249 | R13: | 4345 | 4346 | FOR THE 1ST 128 CHARS. TO ACCESS CHARS IN THE SECOND HALF, THE USER |
| F535 E82100 | 4250 | CALL R17 | 4347 | 4348 | MUST INITIALIZE THE VECTOR AT INTERRUPT 17H (LOCATION 0807CH) TO |
| F536 61EE5020 | 4251 | SUB SI, 2000H+80 | 4349 | 4350 | POINT TO THE USER SUPPLIED TABLE OF GRAPHIC IMAGES (8X8 BOXES). |
| F538 61EF5020 | 4252 | SUB SI, 2000H+80 | 4351 | 4352 | FAILURE TO DO SO WILL CAUSE IN STRANGE RESULTS |
| F53E FECC | 4253 | OEC AH | 4353 | 4354 | ASSUME CS:CODE, DS:DATA, ES:DATA |
| F540 75F1 | 4254 | JNZ R13 | 4355 | 4356 | GRAPHICS_WRITE PROC NEAR |
| | 4255 | ;----- FILL IN THE VACATED LINE(S) | 4357 | 4358 | MOV AH, 0 ; ZERO TO HIGH OF CODE POINT |
| F542 | 4256 | R14: | 4359 | 4360 | PUSH AX ; SAVE CODE POINT VALUE |
| F542 8AC7 | 4257 | MOV AL, 8H | 4361 | 4362 | ;----- DETERMINE POSITION IN REGEN BUFFER TO PUT CODE POINTS |
| F544 | 4258 | | 4363 | 4364 | CALL SUB ; FIND LOCATION IN REGEN BUFFER |
| F544 E82700 | 4259 | CALL R18 | 4365 | 4366 | MOV DI, AX ; REGEN POINTER IN DI |
| F547 61EF5020 | 4260 | SUB DI, 2000H+80 | 4367 | 4368 | ;----- DETERMINE REGION TO GET CODE POINTS FROM |
| F54B FE0B | 4261 | OEC BL | 4369 | 4370 | POP AX ; RECOVER CODE POINT |
| F54D 75F5 | 4262 | JNZ R15 | 4371 | 4372 | CHP AL, 08H ; IS IT IN SECOND HALF |
| F54F FC | 4263 | CLO | 4373 | 4374 | JAE SI ; YES |
| F550 E974C | 4264 | JMP VIDEO_RETURN | 4375 | 4376 | ;----- IMAGE IS IN FIRST HALF, CONTAINED IN ROM |
| | 4265 | | 4377 | 4378 | MOV SI, 0FA0H ; OFFSET CRT_CHAR_GEN-OFFSET OF IMAGES |
| F553 | 4266 | R16: | 4379 | 4380 | PUSH CS ; SAVE SEGMENT ON STACK |
| F553 8ADE | 4267 | MOV BL, 0H | 4381 | 4382 | JMP SHORT S2 ; DETERMINE_MODE |
| F555 E8E0 | 4268 | JMP R14 | | | |
| | 4269 | GRAPHICS_READ ENDP | | | |
| | 4270 | ;----- ROUTINE TO MOVE ONE ROW OF INFORMATION | | | |
| | 4271 | | | | |
| | 4272 | | | | |
| | 4273 | | | | |
| | 4274 | | | | |
| F557 | 4275 | R17 | | | |
| F557 0A0A | 4276 | PROC NEAR | | | |
| F559 56 | 4277 | MOV CL, DL | | | |
| F55A 57 | 4278 | PUSH SI | | | |
| F55B F3 | 4279 | PUSH DI | | | |
| F55C 44 | 4280 | REP MOVSB | | | |
| F55D 5F | 4281 | POP DI | | | |
| F55E 3E | 4282 | POP SI | | | |
| F55F 61C0020 | 4283 | AOD 01, 2000H | | | |
| F563 61C70020 | 4284 | ADD DI, 2000H | | | |
| F567 56 | 4285 | PUSH SI | | | |
| F568 57 | 4286 | PUSH DI | | | |
| F569 0A0A | 4287 | MOV CL, DL | | | |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|-----------------|------|---|------------------------------|------|--|
| F594 | 4359 | ;----- IMAGE IS IN SECOND HALF, IN USER RAM | F607 3C33 | 4635 | AND AX,BX ; CONVERT TO COLOR |
| F594 2C80 | 4360 | SI: | F609 8C280 | 4636 | TEST DL,00H ; AGAIN, IS THIS XOR FUNCTION |
| F594 IE | 4361 | SUB AL,00H ; EXTEND CHAR | JZ DL | 4637 | JZ SI ; NO, JUST STORE THE VALUES |
| F597 20F6 | 4362 | PUSH DS | XOR AX,ES:DI+2000H | 4638 | XOR AX,ES:DI+2000H ; FUNCTION WITH FIRST HALF |
| F599 80E0 | 4363 | SUB SI,SI | XOR AL,ES:DI+2000H | 4639 | XOR AL,ES:DI+2000H ; AND WITH SECOND HALF |
| F59B C5367C00 | 4364 | MOV DS:SI | SI: | | |
| F59F 80D4 | 4365 | ASSUME DS:ABS0 | MOV ES:DI+2000H,AX | 4640 | MOV ES:DI+2000H,AX ; STORE IN SECOND PORTION OF BUFFER |
| F59B C5367C00 | 4366 | LDI SI,EXT_PTR | ADD DI,00 | 4641 | ADD DI,00 ; POINT TO NEXT LOCATION |
| F59F 80D4 | 4367 | MOV DS:SI | DEC DH | 4642 | DEC DH ; KEEP GOING |
| F5A1 IF | 4368 | ASSUME DS:DATA | JNZ S9 | 4643 | JNZ S9 ; RECOVER CODE POINTER |
| F5A2 SE | 4369 | POP DS | POP SI | 4644 | POP SI ; RECOVER REGEN POINTER |
| | 4370 | POP DS | POP DI | 4645 | POP DI ; POINT TO NEXT CHAR POSITION |
| | 4371 | POP DX | ADD DI,2 | 4646 | ADD DI,2 ; MORE TO WRITE |
| | 4372 | POP DX | LOOP S6 | 4647 | LOOP S6 ; VIDEO_RETURN |
| | 4373 | ;----- DETERMINE GRAPHICS MODE IN OPERATION | JMP VIDEO_RETURN | 4648 | JMP VIDEO_RETURN |
| F5A3 D1E0 | 4374 | SI: | GRAPHICS_WRITE ENDP | 4649 | GRAPHICS_WRITE ENDP |
| F5A3 D1E0 | 4375 | SAL AX,1 | ; GRAPHICS_READ | 4650 | ; GRAPHICS_READ |
| F5A7 D1E0 | 4376 | SAL AX,1 | ; GRAPHICS_READ PROC NEAR | 4651 | ; GRAPHICS_READ PROC NEAR |
| F5A9 03F0 | 4377 | SAL AX,1 | GRAPHICS_READ S26 | 4652 | GRAPHICS_READ S26 |
| F5AB 803C490006 | 4378 | ADD SI,AX | CALL S26 | 4653 | CALL S26 ; CONVERTED TO OFFSET IN REGEN |
| F5B0 IF | 4379 | CHP CRT_MODE,6 | MOV SI,AX | 4654 | MOV SI,AX ; SAVE IN SI |
| F5B1 72C | 4380 | POP DS | SUB SP,0 | 4655 | SUB SP,0 ; ALLOCATE SPACE TO SAVE THE READ CODE POINT |
| | 4381 | JC S7 | MOV BP,SP | 4656 | MOV BP,SP ; POINTER TO SAVE AREA |
| | 4382 | JC S7 | | 4657 | |
| | 4383 | ;----- HIGH RESOLUTION MODE | | 4658 | |
| F5B3 | 4384 | S3: | | 4659 | |
| F5B3 57 | 4385 | PUSH DI | ; HIGH CHARS | 4660 | ; HIGH CHARS |
| F5B4 5A | 4386 | PUSH SI | SAVE REGEN POINTER | 4661 | SAVE REGEN POINTER |
| F5B5 8A04 | 4387 | MOV DH,4 | SAVE CODE POINTER | 4662 | SAVE CODE POINTER |
| F5B7 AC | 4388 | MOV DH,4 | NUMBER OF TIMES THROUGH LOOP | 4663 | NUMBER OF TIMES THROUGH LOOP |
| F5B8 F6C3B0 | 4389 | LOOP S8 | | 4664 | |
| F5B8 7516 | 4390 | TEST BL,80H | | 4665 | |
| F5B8 AA | 4391 | JNZ S6 | GET BYTE FROM CODE POINTS | 4666 | GET BYTE FROM CODE POINTS |
| F5B8 AC | 4392 | STOSB | SHOULD HE USE THE FUNCTION | 4667 | SHOULD HE USE THE FUNCTION |
| F5B8 AC | 4393 | STOSB | TO PUT CHAR IN | 4668 | TO PUT CHAR IN |
| F5B8 AC | 4394 | STOSB | STORE IN REGEN BUFFER | 4669 | STORE IN REGEN BUFFER |
| F5B8 AC | 4395 | STOSB | | 4670 | |
| F5B8 AC | 4396 | STOSB | | 4671 | |
| F5B8 AC | 4397 | STOSB | | 4672 | |
| F5B8 AC | 4398 | STOSB | | 4673 | |
| F5B8 AC | 4399 | STOSB | | 4674 | |
| F5B8 AC | 4400 | STOSB | | 4675 | |
| F5B8 AC | 4401 | STOSB | | 4676 | |
| F5B8 AC | 4402 | STOSB | | 4677 | |
| F5B8 AC | 4403 | STOSB | | 4678 | |
| F5B8 AC | 4404 | STOSB | | 4679 | |
| F5B8 AC | 4405 | STOSB | | 4680 | |
| F5B8 AC | 4406 | STOSB | | 4681 | |
| F5B8 AC | 4407 | STOSB | | 4682 | |
| F5B8 AC | 4408 | STOSB | | 4683 | |
| F5B8 AC | 4409 | STOSB | | 4684 | |
| F5B8 AC | 4410 | STOSB | | 4685 | |
| F5B8 AC | 4411 | STOSB | | 4686 | |
| F5B8 AC | 4412 | STOSB | | 4687 | |
| F5B8 AC | 4413 | STOSB | | 4688 | |
| F5B8 AC | 4414 | STOSB | | 4689 | |
| F5B8 AC | 4415 | STOSB | | 4690 | |
| F5B8 AC | 4416 | STOSB | | 4691 | |
| F5B8 AC | 4417 | STOSB | | 4692 | |
| F5B8 AC | 4418 | STOSB | | 4693 | |
| F5B8 AC | 4419 | STOSB | | 4694 | |
| F5B8 AC | 4420 | STOSB | | 4695 | |
| F5B8 AC | 4421 | STOSB | | 4696 | |
| F5B8 AC | 4422 | STOSB | | 4697 | |
| F5B8 AC | 4423 | STOSB | | 4698 | |
| F5B8 AC | 4424 | STOSB | | 4699 | |
| F5B8 AC | 4425 | STOSB | | 4700 | |
| F5B8 AC | 4426 | STOSB | | 4701 | |
| F5B8 AC | 4427 | STOSB | | 4702 | |
| F5B8 AC | 4428 | STOSB | | 4703 | |
| F5B8 AC | 4429 | STOSB | | 4704 | |
| F5B8 AC | 4430 | STOSB | | 4705 | |
| F5B8 AC | 4431 | STOSB | | 4706 | |
| F5B8 AC | 4432 | STOSB | | 4707 | |
| F5B8 AC | 4433 | STOSB | | 4708 | |
| F5B8 AC | 4434 | STOSB | | 4709 | |

| LOC OBJ | LINE | SOURCE |
|---------------|------|---|
| F689 57 | 4511 | PUSH D1 |
| F68A 80800 | 4512 | MOV CX,8 |
| F68D F3 | 4513 | REPZ CMPSB |
| F68E A6 | 4514 | POP DI |
| F68F 5F | 4515 | POP SI |
| F690 5E | 4516 | JZ S10 |
| F691 741E | 4517 | INC AL |
| F693 F6C0 | 4518 | ADD DI,8 |
| F695 83C708 | 4519 | DEC DX |
| F698 44 | 4520 | JNC S17 |
| F699 75D0 | 4521 | ----- CHAR NOT MATCHED, MIGHT BE IN USER SUPPLIED SECOND HALF |
| F69B 3C00 | 4522 | CMP AL,0 |
| F69D 7412 | 4523 | JE S18 |
| F69F 28C0 | 4524 | SUB AX,AX |
| F6A1 8E08 | 4525 | MOV DS:AX |
| F6A3 C43E7C00 | 4526 | ASSUME DS:ASIO |
| F6A7 BCC0 | 4527 | LES DI,EXT_PTR |
| F6A9 08C7 | 4528 | MOV AX,ES |
| F6AB 7404 | 4529 | OR AX,DI |
| F6AD B080 | 4530 | JZ S18 |
| F6AF E802 | 4531 | JMP S16 |
| F6B1 83C408 | 4532 | ASSUME DS:DATA |
| F6B4 E912F8 | 4533 | ----- CHARACTER IS FOUND (AL=0 IF NOT FOUND) |
| F6B7 0E1013 | 4534 | S18: ADD SP,8 |
| F6BA B4C3 | 4535 | JMP VIDEO_RETURN |
| F6BC 51 | 4536 | ----- GRAPHICS_READ ENDP |
| F6BD 800300 | 4537 | ----- EXPAND_RED_COLOR |
| F6C0 08E0 | 4538 | MOV AL,0 |
| F6C2 08E0 | 4539 | MOV AL,1 |
| F6C4 0A0B | 4540 | MOV AL,2 |
| F6C6 E2F8 | 4541 | MOV AL,3 |
| F6C8 B4F8 | 4542 | MOV AL,4 |
| F6CA 59 | 4543 | MOV AL,5 |
| F6CB C3 | 4544 | MOV AL,6 |
| F6CD 08E0 | 4545 | MOV AL,7 |
| F6CF 08E0 | 4546 | MOV AL,8 |
| F6D1 08E0 | 4547 | MOV AL,9 |
| F6D3 08E0 | 4548 | MOV AL,A |
| F6D5 08E0 | 4549 | MOV AL,B |
| F6D7 08E0 | 4550 | MOV AL,C |
| F6D9 08E0 | 4551 | MOV AL,D |
| F6DB 08E0 | 4552 | MOV AL,E |
| F6DD 08E0 | 4553 | MOV AL,F |
| F6DF 08E0 | 4554 | MOV AL,0 |
| F6E1 08E0 | 4555 | MOV AL,1 |
| F6E3 08E0 | 4556 | MOV AL,2 |
| F6E5 08E0 | 4557 | MOV AL,3 |
| F6E7 08E0 | 4558 | MOV AL,4 |
| F6E9 08E0 | 4559 | MOV AL,5 |
| F6EB 08E0 | 4560 | MOV AL,6 |
| F6ED 08E0 | 4561 | MOV AL,7 |
| F6EF 08E0 | 4562 | MOV AL,8 |
| F6F1 08E0 | 4563 | MOV AL,9 |
| F6F3 08E0 | 4564 | MOV AL,A |
| F6F5 08E0 | 4565 | MOV AL,B |
| F6F7 08E0 | 4566 | MOV AL,C |
| F6F9 08E0 | 4567 | MOV AL,D |
| F6FB 08E0 | 4568 | MOV AL,E |
| F6FD 08E0 | 4569 | MOV AL,F |
| F6FF 08E0 | 4570 | MOV AL,0 |
| F701 08E0 | 4571 | MOV AL,1 |
| F703 08E0 | 4572 | MOV AL,2 |
| F705 08E0 | 4573 | MOV AL,3 |
| F707 08E0 | 4574 | MOV AL,4 |
| F709 08E0 | 4575 | MOV AL,5 |
| F70B 08E0 | 4576 | MOV AL,6 |
| F70D 08E0 | 4577 | MOV AL,7 |
| F70F 08E0 | 4578 | MOV AL,8 |
| F711 08E0 | 4579 | MOV AL,9 |
| F713 08E0 | 4580 | MOV AL,A |
| F715 08E0 | 4581 | MOV AL,B |
| F717 08E0 | 4582 | MOV AL,C |
| F719 08E0 | 4583 | MOV AL,D |
| F71B 08E0 | 4584 | MOV AL,E |
| F71D 08E0 | 4585 | MOV AL,F |
| F71F 08E0 | 4586 | MOV AL,0 |
| F721 08E0 | 4587 | MOV AL,1 |
| F723 08E0 | 4588 | MOV AL,2 |
| F725 08E0 | 4589 | MOV AL,3 |
| F727 08E0 | 4590 | MOV AL,4 |
| F729 08E0 | 4591 | MOV AL,5 |
| F72B 08E0 | 4592 | MOV AL,6 |
| F72D 08E0 | 4593 | MOV AL,7 |
| F72F 08E0 | 4594 | MOV AL,8 |
| F731 08E0 | 4595 | MOV AL,9 |
| F733 08E0 | 4596 | MOV AL,A |
| F735 08E0 | 4597 | MOV AL,B |
| F737 08E0 | 4598 | MOV AL,C |
| F739 08E0 | 4599 | MOV AL,D |
| F73B 08E0 | 4600 | MOV AL,E |
| F73D 08E0 | 4601 | MOV AL,F |
| F73F 08E0 | 4602 | MOV AL,0 |
| F741 08E0 | 4603 | MOV AL,1 |
| F743 08E0 | 4604 | MOV AL,2 |
| F745 08E0 | 4605 | MOV AL,3 |
| F747 08E0 | 4606 | MOV AL,4 |
| F749 08E0 | 4607 | MOV AL,5 |
| F74B 08E0 | 4608 | MOV AL,6 |
| F74D 08E0 | 4609 | MOV AL,7 |
| F74F 08E0 | 4610 | MOV AL,8 |
| F751 08E0 | 4611 | MOV AL,9 |
| F753 08E0 | 4612 | MOV AL,A |
| F755 08E0 | 4613 | MOV AL,B |
| F757 08E0 | 4614 | MOV AL,C |
| F759 08E0 | 4615 | MOV AL,D |
| F75B 08E0 | 4616 | MOV AL,E |
| F75D 08E0 | 4617 | MOV AL,F |
| F75F 08E0 | 4618 | MOV AL,0 |
| F761 08E0 | 4619 | MOV AL,1 |
| F763 08E0 | 4620 | MOV AL,2 |
| F765 08E0 | 4621 | MOV AL,3 |
| F767 08E0 | 4622 | MOV AL,4 |
| F769 08E0 | 4623 | MOV AL,5 |
| F76B 08E0 | 4624 | MOV AL,6 |
| F76D 08E0 | 4625 | MOV AL,7 |
| F76F 08E0 | 4626 | MOV AL,8 |
| F771 08E0 | 4627 | MOV AL,9 |
| F773 08E0 | 4628 | MOV AL,A |
| F775 08E0 | 4629 | MOV AL,B |
| F777 08E0 | 4630 | MOV AL,C |
| F779 08E0 | 4631 | MOV AL,D |
| F77B 08E0 | 4632 | MOV AL,E |
| F77D 08E0 | 4633 | MOV AL,F |
| F77F 08E0 | 4634 | MOV AL,0 |
| F781 08E0 | 4635 | MOV AL,1 |
| F783 08E0 | 4636 | MOV AL,2 |
| F785 08E0 | 4637 | MOV AL,3 |
| F787 08E0 | 4638 | MOV AL,4 |
| F789 08E0 | 4639 | MOV AL,5 |
| F78B 08E0 | 4640 | MOV AL,6 |
| F78D 08E0 | 4641 | MOV AL,7 |
| F78F 08E0 | 4642 | MOV AL,8 |
| F791 08E0 | 4643 | MOV AL,9 |
| F793 08E0 | 4644 | MOV AL,A |
| F795 08E0 | 4645 | MOV AL,B |
| F797 08E0 | 4646 | MOV AL,C |
| F799 08E0 | 4647 | MOV AL,D |
| F79B 08E0 | 4648 | MOV AL,E |
| F79D 08E0 | 4649 | MOV AL,F |
| F79F 08E0 | 4650 | MOV AL,0 |
| F7A1 08E0 | 4651 | MOV AL,1 |
| F7A3 08E0 | 4652 | MOV AL,2 |
| F7A5 08E0 | 4653 | MOV AL,3 |
| F7A7 08E0 | 4654 | MOV AL,4 |
| F7A9 08E0 | 4655 | MOV AL,5 |
| F7AB 08E0 | 4656 | MOV AL,6 |
| F7AD 08E0 | 4657 | MOV AL,7 |
| F7AF 08E0 | 4658 | MOV AL,8 |
| F7B1 08E0 | 4659 | MOV AL,9 |
| F7B3 08E0 | 4660 | MOV AL,A |
| F7B5 08E0 | 4661 | MOV AL,B |
| F7B7 08E0 | 4662 | MOV AL,C |
| F7B9 08E0 | 4663 | MOV AL,D |
| F7BB 08E0 | 4664 | MOV AL,E |
| F7BD 08E0 | 4665 | MOV AL,F |
| F7BF 08E0 | 4666 | MOV AL,0 |
| F7C1 08E0 | 4667 | MOV AL,1 |
| F7C3 08E0 | 4668 | MOV AL,2 |
| F7C5 08E0 | 4669 | MOV AL,3 |
| F7C7 08E0 | 4670 | MOV AL,4 |
| F7C9 08E0 | 4671 | MOV AL,5 |
| F7CB 08E0 | 4672 | MOV AL,6 |
| F7CD 08E0 | 4673 | MOV AL,7 |
| F7CF 08E0 | 4674 | MOV AL,8 |
| F7D1 08E0 | 4675 | MOV AL,9 |
| F7D3 08E0 | 4676 | MOV AL,A |
| F7D5 08E0 | 4677 | MOV AL,B |
| F7D7 08E0 | 4678 | MOV AL,C |
| F7D9 08E0 | 4679 | MOV AL,D |
| F7DB 08E0 | 4680 | MOV AL,E |
| F7DD 08E0 | 4681 | MOV AL,F |
| F7DF 08E0 | 4682 | MOV AL,0 |
| F7E1 08E0 | 4683 | MOV AL,1 |
| F7E3 08E0 | 4684 | MOV AL,2 |
| F7E5 08E0 | 4685 | MOV AL,3 |
| F7E7 08E0 | 4686 | MOV AL,4 |
| F7E9 08E0 | 4687 | MOV AL,5 |
| F7EB 08E0 | 4688 | MOV AL,6 |
| F7ED 08E0 | 4689 | MOV AL,7 |
| F7EF 08E0 | 4690 | MOV AL,8 |
| F7F1 08E0 | 4691 | MOV AL,9 |
| F7F3 08E0 | 4692 | MOV AL,A |
| F7F5 08E0 | 4693 | MOV AL,B |
| F7F7 08E0 | 4694 | MOV AL,C |
| F7F9 08E0 | 4695 | MOV AL,D |
| F7FB 08E0 | 4696 | MOV AL,E |
| F7FD 08E0 | 4697 | MOV AL,F |
| F7FF 08E0 | 4698 | MOV AL,0 |
| F801 08E0 | 4699 | MOV AL,1 |
| F803 08E0 | 4700 | MOV AL,2 |
| F805 08E0 | 4701 | MOV AL,3 |
| F807 08E0 | 4702 | MOV AL,4 |
| F809 08E0 | 4703 | MOV AL,5 |
| F80B 08E0 | 4704 | MOV AL,6 |
| F80D 08E0 | 4705 | MOV AL,7 |
| F80F 08E0 | 4706 | MOV AL,8 |
| F811 08E0 | 4707 | MOV AL,9 |
| F813 08E0 | 4708 | MOV AL,A |
| F815 08E0 | 4709 | MOV AL,B |
| F817 08E0 | 4710 | MOV AL,C |
| F819 08E0 | 4711 | MOV AL,D |
| F81B 08E0 | 4712 | MOV AL,E |
| F81D 08E0 | 4713 | MOV AL,F |
| F81F 08E0 | 4714 | MOV AL,0 |
| F821 08E0 | 4715 | MOV AL,1 |
| F823 08E0 | 4716 | MOV AL,2 |
| F825 08E0 | 4717 | MOV AL,3 |
| F827 08E0 | 4718 | MOV AL,4 |
| F829 08E0 | 4719 | MOV AL,5 |
| F82B 08E0 | 4720 | MOV AL,6 |
| F82D 08E0 | 4721 | MOV AL,7 |
| F82F 08E0 | 4722 | MOV AL,8 |
| F831 08E0 | 4723 | MOV AL,9 |
| F833 08E0 | 4724 | MOV AL,A |
| F835 08E0 | 4725 | MOV AL,B |
| F837 08E0 | 4726 | MOV AL,C |
| F839 08E0 | 4727 | MOV AL,D |
| F83B 08E0 | 4728 | MOV AL,E |
| F83D 08E0 | 4729 | MOV AL,F |
| F83F 08E0 | 4730 | MOV AL,0 |
| F841 08E0 | 4731 | MOV AL,1 |
| F843 08E0 | 4732 | MOV AL,2 |
| F845 08E0 | 4733 | MOV AL,3 |
| F847 08E0 | 4734 | MOV AL,4 |
| F849 08E0 | 4735 | MOV AL,5 |
| F84B 08E0 | 4736 | MOV AL,6 |
| F84D 08E0 | 4737 | MOV AL,7 |
| F84F 08E0 | 4738 | MOV AL,8 |
| F851 08E0 | 4739 | MOV AL,9 |
| F853 08E0 | 4740 | MOV AL,A |
| F855 08E0 | 4741 | MOV AL,B |
| F857 08E0 | 4742 | MOV AL,C |
| F859 08E0 | 4743 | MOV AL,D |
| F85B 08E0 | 4744 | MOV AL,E |
| F85D 08E0 | 4745 | MOV AL,F |
| F85F 08E0 | 4746 | MOV AL,0 |
| F861 08E0 | 4747 | MOV AL,1 |
| F863 08E0 | 4748 | MOV AL,2 |
| F865 08E0 | 4749 | MOV AL,3 |
| F867 08E0 | 4750 | MOV AL,4 |
| F869 08E0 | 4751 | MOV AL,5 |
| F86B 08E0 | 4752 | MOV AL,6 |
| F86D 08E0 | 4753 | MOV AL,7 |
| F86F 08E0 | 4754 | MOV AL,8 |
| F871 08E0 | 4755 | MOV AL,9 |
| F873 08E0 | 4756 | MOV AL,A |
| F875 08E0 | 4757 | MOV AL,B |
| F877 08E0 | 4758 | MOV AL,C |
| F879 08E0 | 4759 | MOV AL,D |
| F87B 08E0 | 4760 | MOV AL,E |
| F87D 08E0 | 4761 | MOV AL,F |
| F87F 08E0 | 4762 | MOV AL,0 |
| F881 08E0 | 4763 | MOV AL,1 |
| F883 08E0 | 4764 | MOV AL,2 |
| F885 08E0 | 4765 | MOV AL,3 |
| F887 08E0 | 4766 | MOV AL,4 |
| F889 08E0 | 4767 | MOV AL,5 |
| F88B 08E0 | 4768 | MOV AL,6 |
| F88D 08E0 | 4769 | MOV AL,7 |
| F88F 08E0 | 4770 | MOV AL,8 |
| F891 08E0 | 4771 | MOV AL,9 |
| F893 08E0 | 4772 | MOV AL,A |
| F895 08E0 | 4773 | MOV AL,B |
| F897 08E0 | 4774 | MOV AL,C |
| F899 08E0 | 4775 | MOV AL,D |
| F89B 08E0 | 4776 | MOV AL,E |
| F89D 08E0 | 4777 | MOV AL,F |
| F89F 08E0 | 4778 | MOV AL,0 |
| F8A1 08E0 | 4779 | MOV AL,1 |
| F8A3 08E0 | 4780 | MOV AL,2 |
| F8A5 08E0 | 4781 | MOV AL,3 |
| F8A7 08E0 | 4782 | MOV AL,4 |
| F8A9 08E0 | 4783 | MOV AL,5 |
| F8AB 08E0 | 4784 | MOV AL,6 |
| F8AD 08E0 | 4785 | MOV AL,7 |
| F8AF 08E0 | 4786 | MOV AL,8 |
| F8B1 08E0 | 4787 | MOV AL,9 |
| F8B3 08E0 | 4788 | MOV AL,A |
| F8B5 08E0 | 4789 | MOV AL,B |
| F8B7 08E0 | 4790 | MOV AL,C |
| F8B9 08E0 | 4791 | MOV AL,D |
| F8BB 08E0 | 4792 | MOV AL,E |
| F8BD 08E0 | 4793 | MOV AL,F |
| F8BF 08E0 | 4794 | MOV AL,0 |
| F8C1 08E0 | 4795 | MOV AL,1 |
| F8C3 08E0 | 4796 | MOV AL,2 |
| F8C5 08E0 | 4797 | MOV AL,3 |
| F8C7 08E0 | 4798 | MOV AL,4 |
| F8C9 08E0 | 4799 | MOV AL,5 |
| F8CB 08E0 | 4800 | MOV AL,6 |
| F8CD 08E0 | 4801 | MOV AL,7 |
| F8CF 08E0 | 4802 | MOV AL,8 |
| F8D1 08E0 | 4803 | MOV AL,9 |
| F8D3 08E0 | 4804 | MOV AL,A |
| F8D5 08E0 | 4805 | MOV AL,B |
| F8D7 08E0 | 4806 | MOV AL,C |
| F8D9 08E0 | 4807 | MOV AL,D |
| F8DB 08E0 | 4808 | MOV AL,E |
| F8DD 08E0 | 4809 | MOV AL,F |
| F8DF 08E0 | 4810 | MOV AL,0 |
| F8E1 08E0 | 4811 | MOV AL,1 |
| F8E3 08E0 | 4812 | MOV AL,2 |
| F8E5 08E0 | 4813 | MOV AL,3 |
| F8E7 08E0 | 4814 | MOV AL,4 |
| F8E9 08E0 | 4815 | MOV AL,5 |
| F8EB 08E0 | 4816 | MOV AL,6 |
| F8ED 08E0 | 4817 | MOV AL,7 |
| F8EF 08E0 | 4818 | MOV AL,8 |
| F8F1 08E0 | 4819 | MOV AL,9 |
| F8F3 08E0 | 4820 | MOV AL,A |
| F8F5 08E0 | 4821 | MOV AL,B |
| F8F7 08E0 | 4822 | MOV AL,C |
| F8F9 08E0 | 4823 | MOV AL,D |
| F8FB 08E0 | 4824 | MOV AL,E |
| F8FD 08E0 | 4825 | MOV AL,F |
| F8FF 08E0 | 4826 | MOV AL,0 |
| F901 08E0 | 4827 | MOV AL,1 |
| F903 08E0 | 4828 | MOV AL,2 |
| F905 08E0 | 4829 | MOV AL,3 |
| F907 08E0 | 4830 | MOV AL,4 |
| F909 08E0 | 4831 | MOV AL,5 |
| F90B 08E0 | 4832 | MOV AL,6 |
| F90D 08E0 | 4833 | MOV AL,7 |
| F90F 08E0 | 4834 | MOV AL,8 |
| F911 08E0 | 4835 | MOV AL,9 |
| F913 08E0 | 4836 | MOV AL,A |
| F915 08E0 | 4837 | MOV AL,B |
| F917 08E0 | 4838 | MOV AL,C |
| F919 08E0 | 4839 | MOV AL,D |
| F91B 08E0 | 4840 | MOV AL,E |
| F91D 08E0 | 4841 | MOV AL,F |
| F91F 08E0 | 4842 | MOV AL,0 |
| F921 08E0 | 4843 | MOV AL,1 |
| F923 08E0 | 4844 | MOV AL,2 |
| F925 08E0 | 4845 | MOV AL,3 |
| F927 08E0 | 4846 | MOV AL,4 |
| F929 08E0 | 4847 | MOV AL,5</ |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|---------|------|--|---------|------|-------------------------------------|
| | 4662 | 1 THE 0 COLOR IS USED. | | 4736 | JMP VIDEO_RETURN ; RETURN TO CALLER |
| | 4663 | 1 ENTRY -- | | 4739 | UM: ; SET-CURSOR-INC |
| | 4664 | 1 (AL) = CHARACTER TO BE WRITTEN | | 4740 | IMC DH ; SET-CURSOR |
| | 4665 | 1 NOTE THAT BACK SPACE, CAP RET, BELL AND LINE FEED ARE HANDLED | | 4741 | U7: ; SET-CURSOR |
| | 4666 | 1 AS COMPARES RATHER THAN AS DISPLAYABLE GRAPHICS | | 4742 | MOV AH,2 ; ESTABLISH THE NEW CURSOR |
| | 4667 | 1 (BL) = FOREGROUND COLOR FOR CHAR WRITE IF CURRENTLY IN A GRAPHICS MODE | | 4743 | JMP UN |
| | 4668 | 1 EXIT -- | | 4744 | BACK SPACE FOUND |
| | 4669 | 1 ALL REGISTERS SAVED | | 4745 | UB: ; ALREADY AT END OF LINE |
| | 4670 | 1 ASSUME CS:CODE,DS:DATA | | 4746 | JE U7 ; SET-CURSOR |
| | 4671 | 1 WRITE_TTY PROC NEAR | | 4747 | NO -- JUST MOVE IT BACK |
| | 4672 | 1 PUSH AX ; SAVE REGISTERS | | 4748 | DEC U7 ; SET-CURSOR |
| | 4673 | 1 PUSH AX ; SAVE CHAR TO WRITE | | 4749 | NO -- JUST MOVE IT BACK |
| | 4674 | 1 MOV AH,3 ; READ THE CURRENT CURSOR POSITION | | 4750 | DEC U7 ; SET-CURSOR |
| | 4675 | 1 INT 10H ; RECOVER CHAR | | 4751 | NO -- JUST MOVE IT BACK |
| | 4676 | 1 POP AX ; RECOVER CHAR | | 4752 | DEC U7 ; SET-CURSOR |
| | 4677 | 1 POP AX ; RECOVER CHAR | | 4753 | NO -- JUST MOVE IT BACK |
| | 4678 | 1 POP AX ; RECOVER CHAR | | 4754 | NO -- JUST MOVE IT BACK |
| | 4679 | 1 POP AX ; RECOVER CHAR | | 4755 | NO -- JUST MOVE IT BACK |
| | 4680 | 1 POP AX ; RECOVER CHAR | | 4756 | NO -- JUST MOVE IT BACK |
| | 4681 | 1 POP AX ; RECOVER CHAR | | 4757 | NO -- JUST MOVE IT BACK |
| | 4682 | 1 POP AX ; RECOVER CHAR | | 4758 | NO -- JUST MOVE IT BACK |
| | 4683 | 1 POP AX ; RECOVER CHAR | | 4759 | NO -- JUST MOVE IT BACK |
| | 4684 | 1 POP AX ; RECOVER CHAR | | 4760 | NO -- JUST MOVE IT BACK |
| | 4685 | 1 POP AX ; RECOVER CHAR | | 4761 | NO -- JUST MOVE IT BACK |
| | 4686 | 1 POP AX ; RECOVER CHAR | | 4762 | NO -- JUST MOVE IT BACK |
| | 4687 | 1 POP AX ; RECOVER CHAR | | 4763 | NO -- JUST MOVE IT BACK |
| | 4688 | 1 POP AX ; RECOVER CHAR | | 4764 | NO -- JUST MOVE IT BACK |
| | 4689 | 1 POP AX ; RECOVER CHAR | | 4765 | NO -- JUST MOVE IT BACK |
| | 4690 | 1 POP AX ; RECOVER CHAR | | 4766 | NO -- JUST MOVE IT BACK |
| | 4691 | 1 POP AX ; RECOVER CHAR | | 4767 | NO -- JUST MOVE IT BACK |
| | 4692 | 1 POP AX ; RECOVER CHAR | | 4768 | NO -- JUST MOVE IT BACK |
| | 4693 | 1 POP AX ; RECOVER CHAR | | 4769 | NO -- JUST MOVE IT BACK |
| | 4694 | 1 POP AX ; RECOVER CHAR | | 4770 | NO -- JUST MOVE IT BACK |
| | 4695 | 1 POP AX ; RECOVER CHAR | | 4771 | NO -- JUST MOVE IT BACK |
| | 4696 | 1 POP AX ; RECOVER CHAR | | 4772 | NO -- JUST MOVE IT BACK |
| | 4697 | 1 POP AX ; RECOVER CHAR | | 4773 | NO -- JUST MOVE IT BACK |
| | 4698 | 1 POP AX ; RECOVER CHAR | | 4774 | NO -- JUST MOVE IT BACK |
| | 4699 | 1 POP AX ; RECOVER CHAR | | 4775 | NO -- JUST MOVE IT BACK |
| | 4700 | 1 POP AX ; RECOVER CHAR | | 4776 | NO -- JUST MOVE IT BACK |
| | 4701 | 1 POP AX ; RECOVER CHAR | | 4777 | NO -- JUST MOVE IT BACK |
| | 4702 | 1 POP AX ; RECOVER CHAR | | 4778 | NO -- JUST MOVE IT BACK |
| | 4703 | 1 POP AX ; RECOVER CHAR | | 4779 | NO -- JUST MOVE IT BACK |
| | 4704 | 1 POP AX ; RECOVER CHAR | | 4780 | NO -- JUST MOVE IT BACK |
| | 4705 | 1 POP AX ; RECOVER CHAR | | 4781 | NO -- JUST MOVE IT BACK |
| | 4706 | 1 POP AX ; RECOVER CHAR | | 4782 | NO -- JUST MOVE IT BACK |
| | 4707 | 1 POP AX ; RECOVER CHAR | | 4783 | NO -- JUST MOVE IT BACK |
| | 4708 | 1 POP AX ; RECOVER CHAR | | 4784 | NO -- JUST MOVE IT BACK |
| | 4709 | 1 POP AX ; RECOVER CHAR | | 4785 | NO -- JUST MOVE IT BACK |
| | 4710 | 1 POP AX ; RECOVER CHAR | | 4786 | NO -- JUST MOVE IT BACK |
| | 4711 | 1 POP AX ; RECOVER CHAR | | 4787 | NO -- JUST MOVE IT BACK |
| | 4712 | 1 POP AX ; RECOVER CHAR | | 4788 | NO -- JUST MOVE IT BACK |
| | 4713 | 1 POP AX ; RECOVER CHAR | | 4789 | NO -- JUST MOVE IT BACK |
| | 4714 | 1 POP AX ; RECOVER CHAR | | 4790 | NO -- JUST MOVE IT BACK |
| | 4715 | 1 POP AX ; RECOVER CHAR | | 4791 | NO -- JUST MOVE IT BACK |
| | 4716 | 1 POP AX ; RECOVER CHAR | | 4792 | NO -- JUST MOVE IT BACK |
| | 4717 | 1 POP AX ; RECOVER CHAR | | 4793 | NO -- JUST MOVE IT BACK |
| | 4718 | 1 POP AX ; RECOVER CHAR | | 4794 | NO -- JUST MOVE IT BACK |
| | 4719 | 1 POP AX ; RECOVER CHAR | | 4795 | NO -- JUST MOVE IT BACK |
| | 4720 | 1 POP AX ; RECOVER CHAR | | 4796 | NO -- JUST MOVE IT BACK |
| | 4721 | 1 POP AX ; RECOVER CHAR | | 4797 | NO -- JUST MOVE IT BACK |
| | 4722 | 1 POP AX ; RECOVER CHAR | | 4798 | NO -- JUST MOVE IT BACK |
| | 4723 | 1 POP AX ; RECOVER CHAR | | 4799 | NO -- JUST MOVE IT BACK |
| | 4724 | 1 POP AX ; RECOVER CHAR | | 4800 | NO -- JUST MOVE IT BACK |
| | 4725 | 1 POP AX ; RECOVER CHAR | | 4801 | NO -- JUST MOVE IT BACK |
| | 4726 | 1 POP AX ; RECOVER CHAR | | 4802 | NO -- JUST MOVE IT BACK |
| | 4727 | 1 POP AX ; RECOVER CHAR | | 4803 | NO -- JUST MOVE IT BACK |
| | 4728 | 1 POP AX ; RECOVER CHAR | | 4804 | NO -- JUST MOVE IT BACK |
| | 4729 | 1 POP AX ; RECOVER CHAR | | 4805 | NO -- JUST MOVE IT BACK |
| | 4730 | 1 POP AX ; RECOVER CHAR | | 4806 | NO -- JUST MOVE IT BACK |
| | 4731 | 1 POP AX ; RECOVER CHAR | | 4807 | NO -- JUST MOVE IT BACK |
| | 4732 | 1 POP AX ; RECOVER CHAR | | 4808 | NO -- JUST MOVE IT BACK |
| | 4733 | 1 POP AX ; RECOVER CHAR | | 4809 | NO -- JUST MOVE IT BACK |
| | 4734 | 1 POP AX ; RECOVER CHAR | | 4810 | NO -- JUST MOVE IT BACK |
| | 4735 | 1 POP AX ; RECOVER CHAR | | 4811 | NO -- JUST MOVE IT BACK |
| | 4736 | 1 POP AX ; RECOVER CHAR | | 4812 | NO -- JUST MOVE IT BACK |
| | 4737 | 1 POP AX ; RECOVER CHAR | | 4813 | NO -- JUST MOVE IT BACK |
| | 4738 | 1 POP AX ; RECOVER CHAR | | 4814 | NO -- JUST MOVE IT BACK |

| LOC OBJ | LINE | SOURCE |
|----------------|------|---|
| F7C1 8AC4 | 4815 | MOV AL, AH ; REGISTER TO READ |
| F7C2 EE | 4816 | OUT DX, AL ; SET IT UP |
| F7C3 42 | 4817 | INC DX ; DATA REGISTER |
| F7C3 EC | 4818 | IN CH, DX ; GET THE VALUE |
| F7C4 0A08 | 4819 | MOV AL, DX ; SAVE IN CX |
| F7C4 4A | 4820 | DEC DX ; ADDRESS REGISTER |
| F7C5 FFC4 | 4821 | INC AH ; SECOND DATA REGISTER |
| F7C6 0AC4 | 4822 | MOV AL, AH ; POINT TO DATA REGISTER |
| F7C6 EC | 4823 | OUT DX, AL ; GET SECOND DATA VALUE |
| F7C7 42 | 4824 | INC DX ; AX HAS INPUT VALUE |
| F7C7 EC | 4825 | IN AL, DX ; AX HAS INPUT VALUE |
| F7D0 8A05 | 4826 | MOV AH, CH ; AX HAS THE VALUE READ IN FROM THE 6845 |
| F7D2 8A05000 | 4829 | MOV BL, CRT_MODE ; MODE VALUE TO BX |
| F7D6 2AFF | 4830 | SUB BH, BH ; DETERMINE AMOUNT TO SUBTRACT |
| F7D8 2E2A9A1F7 | 4831 | SUB BH, BH ; DETERMINE AMOUNT TO SUBTRACT |
| F7D0 28C3 | 4832 | SUB AX, BH ; TAKE IT AWAY |
| F7D0 28C3 | 4833 | SUB AX, CRT_START ; CONVERT TO CORRECT PAGE ORIGIN |
| F7D0 28C3 | 4834 | SUB AX, CRT_START ; IF POSITIVE, DETERMINE MODE |
| F7E3 7903 | 4835 | JHS V2 ; <0 PLAYS AS |
| F7E5 800000 | 4836 | MOV AX, 0 ; DETERMINE MODE OF OPERATION |
| F7E5 800000 | 4837 | MOV AX, 0 ; DETERMINE MODE |
| F7E5 800000 | 4838 | MOV AX, 0 ; SET #8 SHIFT COUNT |
| F7E5 800000 | 4839 | MOV AX, 0 ; DETERMINE IF GRAPHICS OR ALPHA |
| F7E5 800000 | 4840 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4841 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4842 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4843 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4844 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4845 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4846 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4847 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4848 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4849 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4850 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4851 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4852 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4853 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4854 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4855 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4856 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4857 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4858 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4859 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4860 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4861 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4862 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4863 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4864 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4865 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4866 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4867 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4868 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4869 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4870 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4871 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4872 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4873 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4874 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4875 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4876 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4877 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4878 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4879 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4880 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4881 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4882 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4883 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4884 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4885 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4886 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4887 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4888 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4889 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4890 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4891 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4892 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4893 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4894 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4895 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4896 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4897 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4898 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4899 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4900 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4901 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4902 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4903 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4904 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4905 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4906 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4907 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4908 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4909 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4910 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4911 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4912 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4913 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4914 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4915 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4916 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4917 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4918 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4919 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4920 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4921 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4922 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4923 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4924 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4925 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4926 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4927 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4928 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4929 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4930 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4931 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4932 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4933 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4934 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4935 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4936 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4937 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4938 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4939 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4940 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4941 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4942 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4943 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4944 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4945 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4946 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4947 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4948 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4949 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4950 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4951 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4952 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4953 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4954 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4955 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4956 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4957 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4958 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4959 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4960 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4961 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4962 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4963 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4964 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4965 | MOV AX, 0 ; ALPHA_PEN |
| F7E5 800000 | 4966 | MOV AX, 0 ; ALPHA_PEN |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|-----------------|------|---|----------------|------|--|
| F840 | 4967 | EQUIPMENT | F805 | 5003 | MOTOR_ON PROC NEAR |
| F840 F8 | 4968 | STI | | 5004 | ----- |
| F840 IE | 4969 | PUSH DS | | 5005 | 1 PURPOSE: |
| F84F B84000 | 4970 | MOV AX,DATA | | 5006 | 1 TO TURN ON CASSETTE MOTOR |
| F852 B808 | 4971 | MOV DS,AX | | 5007 | ----- |
| F854 A11000 | 4972 | MOV AX,EQUIP_FLAG | F805 E461 | 5008 | IN AL,PORT_B |
| F857 IF | 4973 | POP DS | F807 2407 | 5009 | AIO AL,HOT_00H |
| F858 CF | 4974 | IRET | F809 E661 | 5010 | PORT_D_AL |
| | 4975 | EQUIPMENT | F80B 2A64 | 5011 | SUB AH,AM |
| | 4976 | ----- INT 15 ----- | F80D C3 | 5012 | RET |
| | 4977 | 1 CASSETTE I/O | | 5013 | MOTOR_ON ENDP |
| | 4978 | 1 (AH) = 0 TURN CASSETTE MOTOR ON | | 5014 | ----- |
| | 4979 | 1 (AH) = 1 TURN CASSETTE MOTOR OFF | F80E | 5015 | MOTOR_OFF PROC NEAR |
| | 4980 | 1 (AH) = 2 READ 1 OR MORE 256 BYTE BLOCKS FROM CASSETTE | | 5016 | ----- |
| | 4981 | 1 (ES:BX) = POINTER TO DATA BUFFER | | 5017 | 1 PURPOSE: |
| | 4982 | 1 (CX) = COUNT OF BYTES TO READ | | 5018 | 1 TO TURN CASSETTE MOTOR OFF |
| | 4983 | 1 ON EXIT: | | 5019 | ----- |
| | 4984 | 1 (ES:BX) = POINTER TO LAST BYTE READ + 1 | F80E E461 | 5020 | IN AL,PORT_B |
| | 4985 | 1 (DX) = COUNT OF BYTES ACTUALLY READ | F80D 0C08 | 5021 | OR AL,00H |
| | 4986 | 1 (CX) = 0 IF NO ERROR OCCURRED | F802 B8F5 | 5022 | JMP M3 |
| | 4987 | 1 (CX) = 1 IF ERROR OCCURRED | | 5023 | MOTOR_OFF ENDP |
| | 4988 | 1 (AH) = ERROR RETURN IF (CX) = 1 | F804 | 5024 | READ_BLOCK PROC NEAR |
| | 4989 | 1 = 01 IF CRC ERROR HAS DETECTED | | 5025 | ----- |
| | 4990 | 1 = 02 IF DATA TRANSITIONS ARE LOST | | 5026 | 1 PURPOSE: |
| | 4991 | 1 = 04 IF NO DATA WAS FOUND | | 5027 | 1 TO READ 1 OR MORE 256 BYTE BLOCKS FROM CASSETTE |
| | 4992 | 1 (AH) = 3 WRITE 1 OR MORE 256 BYTE BLOCKS TO CASSETTE | | 5028 | ----- |
| | 4993 | 1 (ES:BX) = POINTER TO DATA BUFFER | | 5029 | 1 ON ENTRY: |
| | 4994 | 1 (CX) = COUNT OF BYTES TO WRITE | | 5030 | 1 ES IS SEGMENT FOR MEMORY BUFFER (FOR COMPACT CODE) |
| | 4995 | 1 (EX:BX) = POINTER TO LAST BYTE WRITTEN + 1 | | 5031 | 1 BX POINTS TO START OF MEMORY BUFFER |
| | 4996 | 1 (CX) = 0 | | 5032 | 1 CX CONTAINS NUMBER OF BYTES TO READ |
| | 4997 | 1 (AH) = ANY OTHER THAN ABOVE VALUES CAUSES (CX) = 1 | | 5033 | 1 ON EXIT: |
| | 4998 | 1 AND (AH) IS TO BE RETURNED (INVALID COMMAND). | | 5034 | 1 BX POINTS 1 BYTE PAST LAST BYTE PUT IN MEM |
| | 4999 | 1 (CX) = 0 | | 5035 | 1 CX CONTAINS DECREMENTED BYTE COUNT |
| | 5000 | 1 (CX) = 1 | | 5036 | 1 DX CONTAINS NUMBER OF BYTES ACTUALLY READ |
| | 5001 | 1 (CX) = 2 | | 5037 | 1 CARRY FLAG IS CLEAR IF NO ERROR DETECTED |
| | 5002 | 1 (CX) = 3 | | 5038 | 1 CARRY FLAG IS SET IF CRC ERROR DETECTED |
| F859 | 5003 | CASSETTE_IO PROC FAR | | 5039 | ----- |
| F859 F8 | 5004 | STI | F804 53 | 5040 | PUSH BX |
| F85A IE | 5005 | PUSH AX | F805 51 | 5041 | SAVE CX |
| F85B 50 | 5006 | MOV AX,DATA | F805 51 | 5042 | SAVE SI |
| F85C B84000 | 5007 | MOV DS,AX | F807 B0700 | 5043 | PUSH SI |
| F85F B808 | 5008 | MOV AX,0105,BREAK_7FH | F80A 5A | 5044 | MOV SI,7 |
| F861 002671007F | 5009 | POP AX | F80A EDC01 | 5045 | CALL BEGIN_OP |
| F867 B84000 | 5010 | CALL M1 | F80D | 5046 | IN AL,PORT_C |
| F86A IF | 5011 | POP DS | F80D E462 | 5047 | AIO AL,010H |
| F86B C40200 | 5012 | RET 2 | F807 2410 | 5048 | AND AL,010H |
| F86E | 5013 | CASSETTE_IO ENDP | F8A1 A20800 | 5049 | MOV LAST_VAL,AL |
| | 5014 | NEAR | F8A4 B473F | 5050 | MOV DX,16250 |
| | 5015 | 1 PURPOSE: | | 5051 | 1 WAIT_FOR_EDGE |
| | 5016 | 1 TO CALL APPROPRIATE ROUTINE DEPENDING ON REG AH | F8A7 F66710080 | 5052 | TEST BIOS_BREAK, 80H |
| | 5017 | 1 | F8AC 7403 | 5053 | JZ M6 |
| | 5018 | 1 | F8AE E9A00 | 5054 | JMP M17 |
| | 5019 | 1 AH ROUTINE | | 5055 | 1 JUMP IF BREAK KEY HIT |
| | 5020 | 1 | | 5056 | ----- |
| | 5021 | 1 0 MOTOR ON | F8B1 4A | 5057 | DEC DX |
| | 5022 | 1 1 MOTOR OFF | F8B2 7503 | 5058 | JNZ M7 |
| | 5023 | 1 2 READ CASSETTE BLOCK | F8B4 E9A00 | 5059 | JMP M17 |
| | 5024 | 1 3 WRITE CASSETTE BLOCK | | 5060 | 1 JUMP IF NO LEADER FOUND |
| | 5025 | 1 | | 5061 | ----- |
| | 5026 | 1 | | 5062 | 1 IGNORE FIRST EDGE |
| F86E 0464 | 5027 | OR AH,AH | | 5063 | 1 JUMP IF NO EDGE DETECTED |
| F870 7413 | 5028 | JZ MOTOR_ON | | 5064 | 1 CHECK FOR HALF BITS |
| F872 FECC | 5029 | DEC AH | | 5065 | 1 MUST HAVE AT LEAST THIS MANY ONE SIZE |
| F874 7418 | 5030 | JZ MOTOR_OFF | | 5066 | 1 PULSES BEFORE CHECKING FOR SYNC BIT (0) |
| F876 FECC | 5031 | DEC AH | | 5067 | 1 INTERRUPT TASK REGISTERS |
| F878 741A | 5032 | JZ READ_BLOCK | | 5068 | 1 DISABLE TIMER INTERRUPTS |
| F87A FECC | 5033 | DEC AH | | 5069 | 1 SEARCH-OR |
| F87C 7503 | 5034 | JNZ M2 | | 5070 | 1 CHECK FOR BREAK KEY |
| F87E E9701 | 5035 | JMP WRITE_BLOCK | | 5071 | 1 JUMP IF BREAK KEY HIT |
| | 5036 | 1 | | 5072 | 1 JUMP IF NO LEADER FOUND |
| F881 | 5037 | M2: | | 5073 | ----- |
| F881 B440 | 5038 | MOV AH,080H | | 5074 | 1 BIOS_BREAK, 80H |
| F883 F8 | 5039 | STC | | 5075 | 1 JZ M6 |
| F884 C3 | 5040 | RET | | 5076 | 1 JZ M7 |
| | 5041 | 1 | | 5077 | 1 JZ M17 |
| | 5042 | 1 | | 5078 | 1 JZ M17 |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|-----------|------|------------------------------------|-------------|------|--|
| P80A E304 | 5118 | JCKZ W9 | P936 7403 | 5194 | JZ W17 |
| P80C 718F | 5119 | JNC W4 | P938 7402FF | 5195 | JMP W4 |
| P80E E26B | 5120 | LOOP W8 | P938 | 5196 | M17: NO DATA FROM CASSETTE ERROR, I.E. TIMEOUT |
| P80F | 5121 | W9: F1HD-SYNC | | 5197 | ----- |
| P80D 7266 | 5122 | JC W8 | | 5198 | RESTORE REGS |
| | 5123 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5199 | POP CX |
| | 5124 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5200 | POP CX |
| | 5125 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5201 | POP BX |
| | 5126 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5202 | SUB DX,DX |
| | 5127 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5203 | MOV AH,04H |
| | 5128 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5204 | PUSH AX |
| | 5129 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5205 | M18: IN AL, 021H |
| | 5130 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5206 | AND AL, 0FFH-1 |
| | 5131 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5207 | OUT 021H, AL |
| | 5132 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5208 | CALL MOTOR_OFF |
| | 5133 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5209 | POP AX |
| | 5134 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5210 | POP AX |
| | 5135 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5211 | CHP AH,01H |
| | 5136 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5212 | CPC |
| | 5137 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5213 | SET |
| | 5138 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5214 | ENDP |
| | 5139 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5215 | READ_BLOCK |
| | 5140 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5216 | READ_BYTE |
| | 5141 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5217 | PROC READ |
| | 5142 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5218 | TO READ A BYTE FROM CASSETTE |
| | 5143 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5219 | ON EXIT REG AL CONTAINS READ DATA BYTE |
| | 5144 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5220 | ----- |
| | 5145 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5221 | SAVE REGS BX,CX |
| | 5146 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5222 | PUSH BX |
| | 5147 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5223 | PUSH CX |
| | 5148 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5224 | MOV CL,0H |
| | 5149 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5225 | M19: SET BIT COUNTER FOR 8 BITS |
| | 5150 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5226 | BYT-LASH |
| | 5151 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5227 | SAVE CX |
| | 5152 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5228 | ----- |
| | 5153 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5229 | READ DATA BIT FROM CASSETTE |
| | 5154 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5230 | CALL READ_HALF_BIT |
| | 5155 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5231 | CALL W21 |
| | 5156 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5232 | PUSH BX |
| | 5157 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5233 | PUSH BX |
| | 5158 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5234 | CALL READ_HALF_BIT |
| | 5159 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5235 | POP AX |
| | 5160 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5236 | POP AX |
| | 5161 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5237 | JCZ W21 |
| | 5162 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5238 | ENDP |
| | 5163 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5239 | IF CARRY SET IF ONE BIT |
| | 5164 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5240 | CHP BX, 040FH |
| | 5165 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5241 | CPC |
| | 5166 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5242 | LAHF |
| | 5167 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5243 | POP CX |
| | 5168 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5244 | NOTE: |
| | 5169 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5245 | MS BIT OF BYTE IS READ FIRST. |
| | 5170 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5246 | REG CH IS SHIFTED LEFT WITH |
| | 5171 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5247 | CARRY BEING INSERTED INTO LS |
| | 5172 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5248 | 1. BIT OF CH. |
| | 5173 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5249 | 1. AFTER ALL 8 BITS HAVE BEEN |
| | 5174 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5250 | 1. READ, THE MS BIT OF THE DATA BYTE |
| | 5175 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5251 | 1. WILL BE IN THE MS BIT OF REG CH |
| | 5176 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5252 | 1. ROTATE REG CH LEFT WITH CARRY TO |
| | 5177 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5253 | 1. LS BIT OF REG CH |
| | 5178 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5254 | 1. RESUME CARRY FOR CIRC ROUTINE |
| | 5179 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5255 | 1. GENERATE CRC FOR BIT |
| | 5180 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5256 | 1. LOOP TILL ALL 8 BITS OF DATA |
| | 5181 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5257 | 1. ASSEMBLED IN REG CH |
| | 5182 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5258 | 1. BYTE ASH |
| | 5183 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5259 | 1. RETURN DATA BYTE IN REG AL |
| | 5184 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5260 | 1. RD-BIT-EX |
| | 5185 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5261 | 1. RESTORE REGS CX,BX |
| | 5186 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5262 | POP CX |
| | 5187 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5263 | POP BX |
| | 5188 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5264 | IF FINISHED |
| | 5189 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5265 | 1. NO-DATA |
| | 5190 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5266 | 1. RESTORE CX |
| | 5191 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5267 | 1. INDICATE ERROR |
| | 5192 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5268 | POP CX |
| | 5193 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5269 | STC |
| | 5194 | W9: JUMP IF ONE BIT (STILL LEADER) | | 5270 | STC |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|---------------|------|------------------------------------|---------|------|---|
| F07E EBF9 | 5268 | JMP W20 | F03 | 5310 | WRITE 1 OR MORE 256 BYTE BLOCKS TO CASSETTE |
| | 5269 | READ_BYTE ENDP | | 5311 | ON ENTRY: |
| F080 | 5270 | READ_HALF_BIT PROC NEAR | | 5312 | BX POINTS TO MEMORY BUFFER ADDRESS |
| | 5271 | PURPOSE: | | 5313 | CX CONTAINS NUMBER OF BYTES TO WRITE |
| | 5272 | TO COMPUTE TIME TILL NEXT DATA | | 5314 | ON EXIT: |
| | 5273 | TRANSITION (EDGE) | | 5315 | BX POINTS 1 BYTE PAST LAST BYTE WRITTEN TO CASSETTE |
| | 5274 | | | 5316 | CX IS ZERO |
| | 5275 | | | 5317 | MB_BLOCK: |
| | 5276 | ON ENTRY: | | 5318 | CRC_REG.OFFFH |
| | 5277 | EDGE_CNT CONTAINS LAST EDGE COUNT | | 5319 | MOV DX,256 |
| | 5278 | | | 5320 | MOV AL,ES:10X1 |
| | 5279 | ON EXIT: | | 5321 | CALL WRITE_BYTE |
| | 5280 | AX CONTAINS OLD LAST EDGE COUNT | | 5322 | UNLESS COUNTER ADVANCE PERS & DEC COUNT |
| | 5281 | AX CONTAINS PULSE WIDTH (HALF BIT) | | 5323 | INC BUFFER POINTER |
| | 5282 | | | 5324 | DEC BYTE COUNTER |
| F080 B96400 | 5283 | MOV CX,100 | | 5325 | SKIP-ADV |
| F083 0A248000 | 5284 | MOV AH,LAST_VAL | | 5326 | DEC BLOCK CNT |
| F087 | 5285 | | | 5327 | LOOP TILL 256 BYTE BLOCK |
| F087 E462 | 5286 | IN AL,PORT_C | | 5328 | IS WRITTEN TO TAPE |
| F089 2A10 | 5287 | AND AL,010H | | 5329 | WRITE CRC |
| F08B 3AC4 | 5288 | CMP AL,AH | | 5330 | FOR 256 BYTES |
| F08D E1FA | 5289 | LOPNE W22 | | 5331 | MB-BLK |
| F08E A20800 | 5290 | MOV LAST_VAL,AL | | 5332 | READ_BYTE FROM MEM |
| F092 B000 | 5291 | MOV AL,0 | | 5333 | WRITE IT TO CASSETTE |
| F094 E643 | 5292 | OUT TIM_CTL,AL | | 5334 | UNLESS COUNTER ADVANCE PERS & DEC COUNT |
| F096 E440 | 5293 | IN AL,TIMER0 | | 5335 | INC BUFFER POINTER |
| F098 8A00 | 5294 | MOV AH,AL | | 5336 | DEC BYTE COUNTER |
| F09A E440 | 5295 | IN AL,TIMER0 | | 5337 | SKIP-ADV |
| F09C 8A00 | 5296 | XCHG AL,AH | | 5338 | DEC BLOCK CNT |
| F09E 0B16700 | 5297 | MOV BX,EDGE_CNT | | 5339 | LOOP TILL 256 BYTE BLOCK |
| F0A2 2B06 | 5298 | SUB BX,AX | | 5340 | IS WRITTEN TO TAPE |
| F0A4 A36700 | 5299 | MOV EDGE_CNT,AX | | 5341 | WRITE CRC |
| F0A7 C3 | 5300 | RET | | 5342 | FOR 256 BYTES |
| F0A8 | 5301 | READ_HALF_BIT ENDP | | 5343 | MB-BLK |
| | 5302 | WRITE_BLOCK PROC NEAR | | 5344 | AL:ES:10X1 |
| | 5303 | | | 5345 | CALL WRITE_BYTE |
| | 5304 | | | 5346 | UNLESS COUNTER ADVANCE PERS & DEC COUNT |
| | 5305 | | | 5347 | INC BUFFER POINTER |
| | 5306 | | | 5348 | DEC BYTE COUNTER |
| | 5307 | | | 5349 | SKIP-ADV |
| | 5308 | | | 5350 | DEC BLOCK CNT |
| | 5309 | | | 5351 | LOOP TILL 256 BYTE BLOCK |
| | 5310 | | | 5352 | IS WRITTEN TO TAPE |
| | 5311 | | | 5353 | WRITE CRC |
| | 5312 | | | 5354 | FOR 256 BYTES |
| | 5313 | | | 5355 | MB-BLK |
| | 5314 | | | 5356 | AL:ES:10X1 |
| | 5315 | | | 5357 | CALL WRITE_BYTE |
| | 5316 | | | 5358 | UNLESS COUNTER ADVANCE PERS & DEC COUNT |
| | 5317 | | | 5359 | INC BUFFER POINTER |
| | 5318 | | | 5360 | DEC BYTE COUNTER |
| | 5319 | | | 5361 | SKIP-ADV |
| | 5320 | | | 5362 | DEC BLOCK CNT |
| | 5321 | | | 5363 | LOOP TILL 256 BYTE BLOCK |
| | 5322 | | | 5364 | IS WRITTEN TO TAPE |
| | 5323 | | | 5365 | WRITE CRC |
| | 5324 | | | 5366 | FOR 256 BYTES |
| | 5325 | | | 5367 | MB-BLK |
| | 5326 | | | 5368 | AL:ES:10X1 |
| | 5327 | | | 5369 | CALL WRITE_BYTE |
| | 5328 | | | 5370 | UNLESS COUNTER ADVANCE PERS & DEC COUNT |
| | 5329 | | | 5371 | INC BUFFER POINTER |
| | 5330 | | | 5372 | DEC BYTE COUNTER |
| | 5331 | | | 5373 | SKIP-ADV |
| | 5332 | | | 5374 | DEC BLOCK CNT |
| | 5333 | | | 5375 | LOOP TILL 256 BYTE BLOCK |
| | 5334 | | | 5376 | IS WRITTEN TO TAPE |
| | 5335 | | | 5377 | WRITE CRC |
| | 5336 | | | 5378 | FOR 256 BYTES |
| | 5337 | | | 5379 | MB-BLK |
| | 5338 | | | 5380 | AL:ES:10X1 |
| | 5339 | | | 5381 | CALL WRITE_BYTE |
| | 5340 | | | 5382 | UNLESS COUNTER ADVANCE PERS & DEC COUNT |
| | 5341 | | | 5383 | INC BUFFER POINTER |
| | 5342 | | | 5384 | DEC BYTE COUNTER |
| | 5343 | | | 5385 | SKIP-ADV |
| | 5344 | | | 5386 | DEC BLOCK CNT |
| | 5345 | | | 5387 | LOOP TILL 256 BYTE BLOCK |
| | 5346 | | | 5388 | IS WRITTEN TO TAPE |
| | 5347 | | | 5389 | WRITE CRC |
| | 5348 | | | 5390 | FOR 256 BYTES |
| | 5349 | | | 5391 | MB-BLK |
| | 5350 | | | 5392 | AL:ES:10X1 |
| | 5351 | | | 5393 | CALL WRITE_BYTE |
| | 5352 | | | 5394 | UNLESS COUNTER ADVANCE PERS & DEC COUNT |
| | 5353 | | | 5395 | INC BUFFER POINTER |
| | 5354 | | | 5396 | DEC BYTE COUNTER |
| | 5355 | | | 5397 | SKIP-ADV |
| | 5356 | | | 5398 | DEC BLOCK CNT |
| | 5357 | | | 5399 | LOOP TILL 256 BYTE BLOCK |
| | 5358 | | | 5400 | IS WRITTEN TO TAPE |
| | 5359 | | | 5401 | WRITE CRC |
| | 5360 | | | 5402 | FOR 256 BYTES |
| | 5361 | | | 5403 | MB-BLK |
| | 5362 | | | 5404 | AL:ES:10X1 |
| | 5363 | | | 5405 | CALL WRITE_BYTE |
| | 5364 | | | 5406 | UNLESS COUNTER ADVANCE PERS & DEC COUNT |
| | 5365 | | | 5407 | INC BUFFER POINTER |
| | 5366 | | | 5408 | DEC BYTE COUNTER |
| | 5367 | | | 5409 | SKIP-ADV |
| | 5368 | | | 5410 | DEC BLOCK CNT |
| | 5369 | | | 5411 | LOOP TILL 256 BYTE BLOCK |
| | 5370 | | | 5412 | IS WRITTEN TO TAPE |
| | 5371 | | | 5413 | WRITE CRC |
| | 5372 | | | 5414 | FOR 256 BYTES |
| | 5373 | | | 5415 | MB-BLK |
| | 5374 | | | 5416 | AL:ES:10X1 |
| | 5375 | | | 5417 | CALL WRITE_BYTE |
| | 5376 | | | 5418 | UNLESS COUNTER ADVANCE PERS & DEC COUNT |
| | 5377 | | | 5419 | INC BUFFER POINTER |
| | 5378 | | | 5420 | DEC BYTE COUNTER |
| | 5379 | | | 5421 | SKIP-ADV |
| | 5380 | | | 5422 | DEC BLOCK CNT |
| | 5381 | | | 5423 | LOOP TILL 256 BYTE BLOCK |
| | 5382 | | | 5424 | IS WRITTEN TO TAPE |
| | 5383 | | | 5425 | WRITE CRC |
| | 5384 | | | 5426 | FOR 256 BYTES |
| | 5385 | | | 5427 | MB-BLK |
| | 5386 | | | 5428 | AL:ES:10X1 |
| | 5387 | | | 5429 | CALL WRITE_BYTE |
| | 5388 | | | 5430 | UNLESS COUNTER ADVANCE PERS & DEC COUNT |
| | 5389 | | | 5431 | INC BUFFER POINTER |
| | 5390 | | | 5432 | DEC BYTE COUNTER |
| | 5391 | | | 5433 | SKIP-ADV |
| | 5392 | | | 5434 | DEC BLOCK CNT |
| | 5393 | | | 5435 | LOOP TILL 256 BYTE BLOCK |
| | 5394 | | | 5436 | IS WRITTEN TO TAPE |
| | 5395 | | | 5437 | WRITE CRC |
| | 5396 | | | 5438 | FOR 256 BYTES |
| | 5397 | | | 5439 | MB-BLK |
| | 5398 | | | 5440 | AL:ES:10X1 |
| | 5399 | | | 5441 | CALL WRITE_BYTE |
| | 5400 | | | 5442 | UNLESS COUNTER ADVANCE PERS & DEC COUNT |
| | 5401 | | | 5443 | INC BUFFER POINTER |
| | 5402 | | | 5444 | DEC BYTE COUNTER |
| | 5403 | | | 5445 | SKIP-ADV |
| | 5404 | | | 5446 | DEC BLOCK CNT |
| | 5405 | | | 5447 | LOOP TILL 256 BYTE BLOCK |
| | 5406 | | | 5448 | IS WRITTEN TO TAPE |
| | 5407 | | | 5449 | WRITE CRC |
| | 5408 | | | 5450 | FOR 256 BYTES |
| | 5409 | | | 5451 | MB-BLK |
| | 5410 | | | 5452 | AL:ES:10X1 |
| | 5411 | | | 5453 | CALL WRITE_BYTE |
| | 5412 | | | 5454 | UNLESS COUNTER ADVANCE PERS & DEC COUNT |
| | 5413 | | | 5455 | INC BUFFER POINTER |
| | 5414 | | | 5456 | DEC BYTE COUNTER |
| | 5415 | | | 5457 | SKIP-ADV |
| | 5416 | | | 5458 | DEC BLOCK CNT |
| | 5417 | | | 5459 | LOOP TILL 256 BYTE BLOCK |
| | 5418 | | | 5460 | IS WRITTEN TO TAPE |
| | 5419 | | | 5461 | WRITE CRC |
| | 5420 | | | 5462 | FOR 256 BYTES |
| | 5421 | | | 5463 | MB-BLK |
| | 5422 | | | 5464 | AL:ES:10X1 |
| | 5423 | | | 5465 | CALL WRITE_BYTE |
| | 5424 | | | 5466 | UNLESS COUNTER ADVANCE PERS & DEC COUNT |
| | 5425 | | | 5467 | INC BUFFER POINTER |
| | 5426 | | | 5468 | DEC BYTE COUNTER |
| | 5427 | | | 5469 | SKIP-ADV |
| | 5428 | | | 5470 | DEC BLOCK CNT |
| | 5429 | | | 5471 | LOOP TILL 256 BYTE BLOCK |
| | 5430 | | | 5472 | IS WRITTEN TO TAPE |
| | 5431 | | | 5473 | WRITE CRC |
| | 5432 | | | 5474 | FOR 256 BYTES |
| | 5433 | | | 5475 | MB-BLK |
| | 5434 | | | 5476 | AL:ES:10X1 |
| | 5435 | | | 5477 | CALL WRITE_BYTE |
| | 5436 | | | 5478 | UNLESS COUNTER ADVANCE PERS & DEC COUNT |
| | 5437 | | | 5479 | INC BUFFER POINTER |
| | 5438 | | | 5480 | DEC BYTE COUNTER |
| | 5439 | | | 5481 | SKIP-ADV |
| | 5440 | | | 5482 | DEC BLOCK CNT |
| | 5441 | | | 5483 | LOOP TILL 256 BYTE BLOCK |
| | 5442 | | | 5484 | IS WRITTEN TO TAPE |
| | 5443 | | | 5485 | WRITE CRC |
| | 5444 | | | 5486 | FOR 256 BYTES |
| | 5445 | | | 5487 | MB-BLK |
| | 5446 | | | 5488 | AL:ES:10X1 |
| | 5447 | | | 5489 | CALL WRITE_BYTE |
| | 5448 | | | 5490 | UNLESS COUNTER ADVANCE PERS & DEC COUNT |
| | 5449 | | | 5491 | INC BUFFER POINTER |
| | 5450 | | | 5492 | DEC BYTE COUNTER |
| | 5451 | | | 5493 | SKIP-ADV |
| | 5452 | | | 5494 | DEC BLOCK CNT |
| | 5453 | | | 5495 | LOOP TILL 256 BYTE BLOCK |
| | 5454 | | | 5496 | IS WRITTEN TO TAPE |
| | 5455 | | | 5497 | WRITE CRC |
| | 5456 | | | 5498 | FOR 256 BYTES |
| | 5457 | | | 5499 | MB-BLK |
| | 5458 | | | 5500 | AL:ES:10X1 |
| | 5459 | | | 5501 | CALL WRITE_BYTE |
| | 5460 | | | 5502 | UNLESS COUNTER ADVANCE PERS & DEC COUNT |
| | 5461 | | | 5503 | INC BUFFER POINTER |
| | 5462 | | | 5504 | DEC BYTE COUNTER |
| | 5463 | | | 5505 | SKIP-ADV |
| | 5464 | | | 5506 | DEC BLOCK CNT |
| | 5465 | | | 5507 | LOOP TILL 256 BYTE BLOCK |
| | 5466 | | | 5508 | IS WRITTEN TO TAPE |
| | 5467 | | | 5509 | WRITE CRC |
| | 5468 | | | 5510 | FOR 256 BYTES |
| | 5469 | | | 5511 | MB-BLK |
| | 5470 | | | 5512 | AL:ES:10X1 |
| | 5471 | | | 5513 | CALL WRITE_BYTE |
| | 5472 | | | 5514 | UNLESS COUNTER ADVANCE PERS & DEC COUNT |
| | 5473 | | | 5515 | INC BUFFER POINTER |
| | 5474 | | | 5516 | DEC BYTE COUNTER |
| | 5475 | | | 5517 | SKIP-ADV |
| | 5476 | | | 5518 | DEC BLOCK CNT |
| | 5477 | | | 5519 | LOOP TILL 256 BYTE BLOCK |
| | 5478 | | | 5520 | IS WRITTEN TO TAPE |
| | 5479 | | | 5521 | WRITE CRC |
| | 5480 | | | 5522 | FOR 256 BYTES |
| | 5481 | | | 5523 | MB-BLK |
| | 5482 | | | 5524 | AL:ES:10X1 |
| | 5483 | | | 5525 | CALL WRITE_BYTE |
| | 5484 | | | 5526 | UNLESS COUNTER ADVANCE PERS & DEC COUNT |
| | 5485 | | | 5527 | INC BUFFER POINTER |
| | 5486 | | | 5528 | DEC BYTE COUNTER |
| | 5487 | | | 5529 | SKIP-ADV |
| | 5488 | | | 5530 | DEC BLOCK CNT |
| | 5489 | | | 5531 | LOOP TILL 256 BYTE BLOCK |
| | 5490 | | | 5532 | IS WRITTEN TO TAPE |
| | 5491 | | | 5533 | WRITE CRC |
| | 5492 | | | 5534 | FOR 256 BYTES |
| | 5493 | | | 5535 | MB-BLK |
| | 5494 | | | 5536 | AL:ES:10X1 |
| | 5495 | | | 5537 | CALL WRITE_BYTE |
| | 5496 | | | 5538 | UNLESS COUNTER ADVANCE PERS & DEC COUNT |
| | 5497 | | | 5539 | INC BUFFER POINTER |
| | 5498 | | | 5540 | DEC BYTE COUNTER |
| | 5499 | | | 5541 | SKIP-ADV |
| | 5500 | | | 5542 | DEC BLOCK CNT |
| | 5501 | | | 5543 | LOOP TILL 256 BYTE BLOCK |
| | 5502 | | | 5544 | IS WRITTEN TO TAPE |
| | 5503 | | | 5545 | WRITE CRC |
| | 5504 | | | 5546 | FOR 256 BYTES |
| | 5505 | | | 5547 | MB-BLK |
| | 5506 | | | 5548 | AL:ES:10X1 |
| | 5507 | | | 5549 | CALL WRITE_BYTE |
| | 5508 | | | 5550 | UNLESS COUNTER ADVANCE PERS & DEC COUNT |
| | 5509 | | | 5551 | INC BUFFER POINTER |
| | 5510 | | | 5552 | DEC BYTE COUNTER |
| | 5511 | | | 5553 | SKIP-ADV |
| | 5512 | | | 5554 | DEC BLOCK CNT |
| | 5513 | | | 5555 | LOOP TILL 256 BYTE BLOCK |
| | 5514 | | | 5556 | IS WRITTEN TO TAPE |
| | 5515 | | | 5557 | WRITE CRC |
| | 5516 | | | 5558 | FOR 256 BYTES |
| | 5517 | | | 5559 | MB-BLK |
| | 5518 | | | 5560 | AL:ES:10X1 |
| | 5519 | | | 5561 | CALL WRITE_BYTE |
| | 5520 | | | 5562 | UNLESS COUNTER ADVANCE PERS & DEC COUNT |
| | 5521 | | | 5563 | INC BUFFER POINTER |
| | 5522 | | | 5564 | DEC BYTE COUNTER |
| | 5523 | | | 5565 | SKIP-ADV |
| | 5524 | | | 556 | |

| LOC OBJ | LINE | SOURCE |
|---------|------|-------------------------------------|
| FAZD C3 | 5415 | RET |
| FAZD C3 | 5416 | WRITE_BYTE |
| FAZD C3 | 5417 | WRITE_BIT |
| FAZD C3 | 5418 | PROC |
| FAZD C3 | 5419 | HEAR |
| FAZD C3 | 5420 | IMPOSE: |
| FAZD C3 | 5421 | TO WRITE A DATA BIT TO CASSETTE |
| FAZD C3 | 5422 | CARRY FLAG CONTAINS DATA BIT |
| FAZD C3 | 5423 | I.E. IF SET DATA BIT IS A ONE |
| FAZD C3 | 5424 | IF CLEAR DATA BIT IS A ZERO |
| FAZD C3 | 5425 | |
| FAZD C3 | 5426 | NOTE: TWO EDGES ARE WRITTEN PER BIT |
| FAZD C3 | 5427 | ONE BIT HAS 500 USEC BETWEEN EDGES |
| FAZD C3 | 5428 | FOR A 1000 USEC PERIOD (1 MILLISEC) |
| FAZD C3 | 5431 | ZERO BIT HAS 250 USEC BETWEEN EDGES |
| FAZD C3 | 5432 | CARRY FLAG IS DATA BIT |
| FAZD C3 | 5433 | IMPOSE: |
| FAZD C3 | 5434 | TO WRITE A DATA BIT TO CASSETTE |
| FAZD C3 | 5435 | CARRY FLAG CONTAINS DATA BIT |
| FAZD C3 | 5436 | I.E. IF SET DATA BIT IS A ONE |
| FAZD C3 | 5437 | IF CLEAR DATA BIT IS A ZERO |
| FAZD C3 | 5438 | |
| FAZD C3 | 5439 | NOTE: TWO EDGES ARE WRITTEN PER BIT |
| FAZD C3 | 5440 | ONE BIT HAS 500 USEC BETWEEN EDGES |
| FAZD C3 | 5441 | FOR A 1000 USEC PERIOD (1 MILLISEC) |
| FAZD C3 | 5442 | ZERO BIT HAS 250 USEC BETWEEN EDGES |
| FAZD C3 | 5443 | CARRY FLAG IS DATA BIT |
| FAZD C3 | 5444 | IMPOSE: |
| FAZD C3 | 5445 | TO WRITE A DATA BIT TO CASSETTE |
| FAZD C3 | 5446 | CARRY FLAG CONTAINS DATA BIT |
| FAZD C3 | 5447 | I.E. IF SET DATA BIT IS A ONE |
| FAZD C3 | 5448 | IF CLEAR DATA BIT IS A ZERO |
| FAZD C3 | 5449 | |
| FAZD C3 | 5450 | NOTE: TWO EDGES ARE WRITTEN PER BIT |
| FAZD C3 | 5451 | ONE BIT HAS 500 USEC BETWEEN EDGES |
| FAZD C3 | 5452 | FOR A 1000 USEC PERIOD (1 MILLISEC) |
| FAZD C3 | 5453 | ZERO BIT HAS 250 USEC BETWEEN EDGES |
| FAZD C3 | 5454 | CARRY FLAG IS DATA BIT |
| FAZD C3 | 5455 | IMPOSE: |
| FAZD C3 | 5456 | TO WRITE A DATA BIT TO CASSETTE |
| FAZD C3 | 5457 | CARRY FLAG CONTAINS DATA BIT |
| FAZD C3 | 5458 | I.E. IF SET DATA BIT IS A ONE |
| FAZD C3 | 5459 | IF CLEAR DATA BIT IS A ZERO |
| FAZD C3 | 5460 | |
| FAZD C3 | 5461 | NOTE: TWO EDGES ARE WRITTEN PER BIT |
| FAZD C3 | 5462 | ONE BIT HAS 500 USEC BETWEEN EDGES |
| FAZD C3 | 5463 | FOR A 1000 USEC PERIOD (1 MILLISEC) |
| FAZD C3 | 5464 | ZERO BIT HAS 250 USEC BETWEEN EDGES |
| FAZD C3 | 5465 | CARRY FLAG IS DATA BIT |
| FAZD C3 | 5466 | IMPOSE: |
| FAZD C3 | 5467 | TO WRITE A DATA BIT TO CASSETTE |
| FAZD C3 | 5468 | CARRY FLAG CONTAINS DATA BIT |
| FAZD C3 | 5469 | I.E. IF SET DATA BIT IS A ONE |
| FAZD C3 | 5470 | IF CLEAR DATA BIT IS A ZERO |
| FAZD C3 | 5471 | |
| FAZD C3 | 5472 | NOTE: TWO EDGES ARE WRITTEN PER BIT |
| FAZD C3 | 5473 | ONE BIT HAS 500 USEC BETWEEN EDGES |
| FAZD C3 | 5474 | FOR A 1000 USEC PERIOD (1 MILLISEC) |
| FAZD C3 | 5475 | ZERO BIT HAS 250 USEC BETWEEN EDGES |
| FAZD C3 | 5476 | CARRY FLAG IS DATA BIT |
| FAZD C3 | 5477 | IMPOSE: |
| FAZD C3 | 5478 | TO WRITE A DATA BIT TO CASSETTE |
| FAZD C3 | 5479 | CARRY FLAG CONTAINS DATA BIT |
| FAZD C3 | 5480 | I.E. IF SET DATA BIT IS A ONE |
| FAZD C3 | 5481 | IF CLEAR DATA BIT IS A ZERO |
| FAZD C3 | 5482 | |
| FAZD C3 | 5483 | NOTE: TWO EDGES ARE WRITTEN PER BIT |
| FAZD C3 | 5484 | ONE BIT HAS 500 USEC BETWEEN EDGES |
| FAZD C3 | 5485 | FOR A 1000 USEC PERIOD (1 MILLISEC) |
| FAZD C3 | 5486 | ZERO BIT HAS 250 USEC BETWEEN EDGES |
| FAZD C3 | 5487 | CARRY FLAG IS DATA BIT |
| FAZD C3 | 5488 | IMPOSE: |
| FAZD C3 | 5489 | TO WRITE A DATA BIT TO CASSETTE |
| FAZD C3 | 5490 | CARRY FLAG CONTAINS DATA BIT |
| FAZD C3 | 5491 | I.E. IF SET DATA BIT IS A ONE |
| FAZD C3 | 5492 | IF CLEAR DATA BIT IS A ZERO |
| FAZD C3 | 5493 | |
| FAZD C3 | 5494 | NOTE: TWO EDGES ARE WRITTEN PER BIT |
| FAZD C3 | 5495 | ONE BIT HAS 500 USEC BETWEEN EDGES |
| FAZD C3 | 5496 | FOR A 1000 USEC PERIOD (1 MILLISEC) |
| FAZD C3 | 5497 | ZERO BIT HAS 250 USEC BETWEEN EDGES |
| FAZD C3 | 5498 | CARRY FLAG IS DATA BIT |
| FAZD C3 | 5499 | IMPOSE: |
| FAZD C3 | 5500 | TO WRITE A DATA BIT TO CASSETTE |
| FAZD C3 | 5501 | CARRY FLAG CONTAINS DATA BIT |
| FAZD C3 | 5502 | I.E. IF SET DATA BIT IS A ONE |
| FAZD C3 | 5503 | IF CLEAR DATA BIT IS A ZERO |
| FAZD C3 | 5504 | |
| FAZD C3 | 5505 | NOTE: TWO EDGES ARE WRITTEN PER BIT |
| FAZD C3 | 5506 | ONE BIT HAS 500 USEC BETWEEN EDGES |
| FAZD C3 | 5507 | FOR A 1000 USEC PERIOD (1 MILLISEC) |
| FAZD C3 | 5508 | ZERO BIT HAS 250 USEC BETWEEN EDGES |
| FAZD C3 | 5509 | CARRY FLAG IS DATA BIT |
| FAZD C3 | 5510 | IMPOSE: |
| FAZD C3 | 5511 | TO WRITE A DATA BIT TO CASSETTE |
| FAZD C3 | 5512 | CARRY FLAG CONTAINS DATA BIT |
| FAZD C3 | 5513 | I.E. IF SET DATA BIT IS A ONE |
| FAZD C3 | 5514 | IF CLEAR DATA BIT IS A ZERO |
| FAZD C3 | 5515 | |
| FAZD C3 | 5516 | NOTE: TWO EDGES ARE WRITTEN PER BIT |
| FAZD C3 | 5517 | ONE BIT HAS 500 USEC BETWEEN EDGES |
| FAZD C3 | 5518 | FOR A 1000 USEC PERIOD (1 MILLISEC) |
| FAZD C3 | 5519 | ZERO BIT HAS 250 USEC BETWEEN EDGES |
| FAZD C3 | 5520 | CARRY FLAG IS DATA BIT |
| FAZD C3 | 5521 | IMPOSE: |
| FAZD C3 | 5522 | TO WRITE A DATA BIT TO CASSETTE |
| FAZD C3 | 5523 | CARRY FLAG CONTAINS DATA BIT |
| FAZD C3 | 5524 | I.E. IF SET |

| LOC OBJ | LINE | SOURCE | LOC OBJ | LINE | SOURCE |
|-----------------------|------|--------|---|------|---|
| FC64 0000000000000000 | 5560 | DB | 000H,030H,030H,000H,000H,010H,030H,000H,1,0,0,3B | 5641 | 1--- INT 1A ----- |
| FC64 1000000000000000 | 5561 | DB | 010H,070H,060H,000H,000H,030H,010H,000H,1,0,0,3C | 5642 | TIME_OF_DAY |
| FC64 0000000000000000 | 5562 | DB | 000H,000H,000H,000H,000H,000H,000H,000H,1,0,0,3D | 5643 | THIS ROUTINE ALLOWS THE CLOCK TO BE SET/READ |
| FC64 0000000000000000 | 5563 | DB | 000H,030H,010H,000H,010H,030H,000H,000H,1,0,0,3E | 5644 | INPUT |
| FC64 7000000000000000 | 5564 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,3F | 5645 | READ THE CURRENT CLOCK SETTING |
| FC64 7000000000000000 | 5565 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,40 | 5646 | (AM) = 0 |
| FC64 7000000000000000 | 5566 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,41 | 5647 | RETURNS CX = HIGH PORTION OF COUNT |
| FC64 7000000000000000 | 5567 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,42 | 5648 | CX = LOW PORTION OF COUNT |
| FC64 7000000000000000 | 5568 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,43 | 5649 | AL = 0 IF TIMER HAS NOT PASSED 24 HOURS SINCE LAST READ |
| FC64 7000000000000000 | 5569 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,44 | 5650 | IF ON ANOTHER DAY |
| FC64 7000000000000000 | 5570 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,45 | 5651 | SET THE CURRENT CLOCK |
| FC64 7000000000000000 | 5571 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,46 | 5652 | CX = HIGH PORTION OF COUNT |
| FC64 7000000000000000 | 5572 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,47 | 5653 | CX = LOW PORTION OF COUNT |
| FC64 7000000000000000 | 5573 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,48 | 5654 | NOTE: COUNTS OCCUR AT THE RATE OF 1/193100/45536 COUNTS/SEC |
| FC64 7000000000000000 | 5574 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,49 | 5655 | (OR ABOUT 16.2 PER SECOND -- SEE EQUATES BELOW) |
| FC64 7000000000000000 | 5575 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,50 | 5656 | ASSUME CS:CODE:DS:DATA |
| FC64 7000000000000000 | 5576 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,51 | 5657 | TIME_OF_DAY PROC FAR |
| FC64 7000000000000000 | 5577 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,52 | 5658 | STI |
| FC64 7000000000000000 | 5578 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,53 | 5659 | INTERRUPTS BACK ON |
| FC64 7000000000000000 | 5579 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,54 | 5660 | SAVE SEGMENT |
| FC64 7000000000000000 | 5580 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,55 | 5661 | SAVE PARAM |
| FC64 7000000000000000 | 5581 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,56 | 5662 | MOV AX,DATA |
| FC64 7000000000000000 | 5582 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,57 | 5663 | MOV DS,AX |
| FC64 7000000000000000 | 5583 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,58 | 5664 | POP AX |
| FC64 7000000000000000 | 5584 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,59 | 5665 | GET BACK INPUT PARAM |
| FC64 7000000000000000 | 5585 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,60 | 5666 | AH=0 |
| FC64 7000000000000000 | 5586 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,61 | 5667 | READ_TIME |
| FC64 7000000000000000 | 5587 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,62 | 5668 | AH=1 |
| FC64 7000000000000000 | 5588 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,63 | 5669 | SET_TIME |
| FC64 7000000000000000 | 5589 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,64 | 5670 | INTERRUPTS BACK ON |
| FC64 7000000000000000 | 5590 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,65 | 5671 | RECOVER SEGMENT |
| FC64 7000000000000000 | 5591 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,66 | 5672 | RETURN TO CALLER |
| FC64 7000000000000000 | 5592 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,67 | 5673 | READ_TIME |
| FC64 7000000000000000 | 5593 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,68 | 5674 | NO TIMER INTERRUPTS WHILE READING |
| FC64 7000000000000000 | 5594 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,69 | 5675 | AL:TIMER_DFL |
| FC64 7000000000000000 | 5595 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,70 | 5676 | TIMER_DFL_O |
| FC64 7000000000000000 | 5596 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,71 | 5677 | GET OVERFLOW, AND RESET THE FLAG |
| FC64 7000000000000000 | 5597 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,72 | 5678 | CX:TIMER_HIGH |
| FC64 7000000000000000 | 5598 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,73 | 5679 | DX:TIMER_LOW |
| FC64 7000000000000000 | 5599 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,74 | 5680 | NO_TIMER |
| FC64 7000000000000000 | 5600 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,75 | 5681 | TOO_RETURN |
| FC64 7000000000000000 | 5601 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,76 | 5682 | SET_TIME |
| FC64 7000000000000000 | 5602 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,77 | 5683 | NO INTERRUPTS WHILE WRITING |
| FC64 7000000000000000 | 5603 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,78 | 5684 | TIMER_LOW:DX |
| FC64 7000000000000000 | 5604 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,79 | 5685 | NO_TIMER |
| FC64 7000000000000000 | 5605 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,80 | 5686 | SET_TIME |
| FC64 7000000000000000 | 5606 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,81 | 5687 | RESET OVERFLOW |
| FC64 7000000000000000 | 5607 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,82 | 5688 | NO_TIMER |
| FC64 7000000000000000 | 5608 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,83 | 5689 | TOO_RETURN |
| FC64 7000000000000000 | 5609 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,84 | 5690 | END |
| FC64 7000000000000000 | 5610 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,85 | 5691 | THIS ROUTINE HANDLES THE TIMER INTERRUPT FROM |
| FC64 7000000000000000 | 5611 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,86 | 5692 | CHANNEL 0 OF THE 8253 TIMER. INPUT FREQUENCY IS 1.19318 MHZ |
| FC64 7000000000000000 | 5612 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,87 | 5693 | AND THE DIVISOR IS 65536, RESULTING IN APPROX. 16.2 INTERRUPTS |
| FC64 7000000000000000 | 5613 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,88 | 5694 | EVERY SECOND. |
| FC64 7000000000000000 | 5614 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,89 | 5695 | THE INTERRUPT HANDLER MAINTAINS A COUNT OF INTERRUPTS SINCE POWER |
| FC64 7000000000000000 | 5615 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,90 | 5696 | ON TIME, WHICH MAY BE USED TO ESTABLISH TIME OF DAY. |
| FC64 7000000000000000 | 5616 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,91 | 5697 | THE INTERRUPT HANDLER ALSO DECREMENTS THE MOTOR CONTROL COUNT |
| FC64 7000000000000000 | 5617 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,92 | 5698 | OF THE DISKETTE, AND WHEN IT EXPIRES, WILL TURN OFF THE DISKETTE |
| FC64 7000000000000000 | 5618 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,93 | 5699 | MOTOR, AND RESET THE MOTOR RUNNING FLAG |
| FC64 7000000000000000 | 5619 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,94 | 5700 | THE INTERRUPT HANDLER WILL ALSO INVOKE A USER ROUTINE THROUGH INTERRUPT |
| FC64 7000000000000000 | 5620 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,95 | 5701 | ON AT EVERY TIME TICK. THE USER MUST CODE A ROUTINE AND PLACE THE |
| FC64 7000000000000000 | 5621 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,96 | 5702 | CORRECT ADDRESS IN THE VECTOR TABLE. |
| FC64 7000000000000000 | 5622 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,97 | 5703 | ----- |
| FC64 7000000000000000 | 5623 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,98 | 5704 | TIME_OF_DAY PROC FAR |
| FC64 7000000000000000 | 5624 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,99 | 5705 | STI |
| FC64 7000000000000000 | 5625 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,100 | 5706 | INTERRUPTS BACK ON |
| FC64 7000000000000000 | 5626 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,101 | 5707 | PUSH AX |
| FC64 7000000000000000 | 5627 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,102 | 5708 | PUSH DX |
| FC64 7000000000000000 | 5628 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,103 | 5709 | MOV AX,DATA |
| FC64 7000000000000000 | 5629 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,104 | 5710 | MOV DS,AX |
| FC64 7000000000000000 | 5630 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,105 | 5711 | INCREMENT TIME |
| FC64 7000000000000000 | 5631 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,106 | 5712 | JNZ T4 |
| FC64 7000000000000000 | 5632 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,107 | 5713 | INC TIMER_HIGH |
| FC64 7000000000000000 | 5633 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,108 | 5714 | TEST_DAY |
| FC64 7000000000000000 | 5634 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,109 | | |
| FC64 7000000000000000 | 5635 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,110 | | |
| FC64 7000000000000000 | 5636 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,111 | | |
| FC64 7000000000000000 | 5637 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,112 | | |
| FC64 7000000000000000 | 5638 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,113 | | |
| FC64 7000000000000000 | 5639 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,114 | | |
| FC64 7000000000000000 | 5640 | DB | 070H,000H,000H,010H,030H,000H,030H,000H,1,0,0,115 | | |

A-78

Notes for the BIOS Listing

1. The wait loop for the printer times out on form feed of > 51 lines. — line ref (3069)
2. Mode controls for the 320 x 200 video have Color/BW reversed. — line ref (3338)
3. The RS232 Timeout is 80 decimal, not 80 hexadecimal. — line ref (1566)
4. The Base Pointer register is destroyed by some video calls.
5. D,04 ◊ character in the character generator has 08 as its last value, S/80. — line ref (5511)
6. If you hit print screen in the Color/Graphics 80x25 Character Mode, the screen may not display during the print cycle.

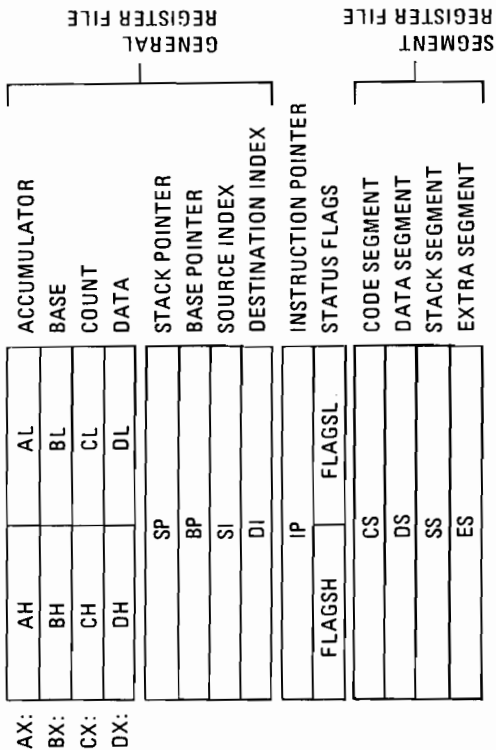
| LOC OBJ | LINE | SOURCE |
|---------------------|------|--|
| | 5865 | ***** |
| | 5867 | 1 THE LOOP FROM PRI10 TO THE INSTRUCTION PRIOR TO PRI20 |
| | 5868 | 1 IS THE LOOP TO READ EACH CURSOR POSITION FROM THE SCREEN |
| | 5869 | 1 AND PRINT. |
| | 5870 | ***** |
| | 5871 | PRI10: MOV AH,2 ;TO INDICATE CURSOR SET REQUEST |
| FF77 B402 | 5872 | INT 10H ;THEN CURSOR POSITION ESTABLISHED |
| FF81 C010 | 5873 | MOV AH,0 ;TO INDICATE READ CHARACTER |
| FF83 B408 | 5874 | INT 10H ;CHARACTER NON IN I/O |
| FF85 C010 | 5875 | OR AL,AL ;SEE IF VALID CHAR |
| FF87 B4C0 | 5876 | JNZ PRI15 ;JUMP IF VALID CHAR |
| FF89 F502 | 5877 | MOV AL,' ' ;MAKE A BLANK |
| FF8B B020 | 5878 | ***** |
| FF8D | 5879 | PRI15: PUSH DX ;SAVE CURSOR POSITION |
| FF8F 52 | 5880 | XOR DX,DX ;INDICATE PRINTER 1 |
| FF91 3302 | 5881 | XOR AH,AH ;TO INDICATE PRINT CHAR IN I/O |
| FF93 32E4 | 5882 | INT 17H ;PRINT THE CHARACTER |
| FF95 5A | 5883 | POP DX ;RECALL CURSOR POSITION |
| FF97 5C425 | 5884 | TEST AH,25H ;TEST FOR PRINTER ERROR |
| FF99 7521 | 5885 | JNZ ERR10 ;JUMP IF ERROR DETECTED |
| FF9B FEC2 | 5886 | INC DL ;ADVANCE TO NEXT COLUMN |
| FF9D 3AC4 | 5887 | CHP CL,DL ;SEE IF AT END OF LINE |
| FF9F 350F | 5888 | JNZ PRI10 ;IF NOT PROCEED |
| FFA1 3202 | 5889 | XOR DL,DL ;BACK TO COLUMN 0 |
| FFA3 B4E2 | 5890 | MOV AH,DL ;[AH]=0 |
| FFA5 52 | 5891 | PUSH DX ;SAVE NEW CURSOR POSITION |
| FFA7 E2700 | 5892 | CALL CRLF ;LINE FEED CARRIAGE RETURN |
| FFA9 5A | 5893 | POP DX ;RECALL CURSOR POSITION |
| FFAB FEC6 | 5894 | INC DL ;ADVANCE TO NEXT LINE |
| FFAD 34EE | 5895 | CHP CL,DL ;FINISHED? |
| FFAF 7500 | 5896 | JNZ PRI20 ;IF NOT CONTINUE |
| FFB1 B402 | 5897 | POP DX ;RECALL CURSOR POSITION |
| FFB3 C010 | 5898 | INT 10H ;TO INDICATE CURSOR SET REQUEST |
| FFB5 C0600000 | 5899 | MOV STATUS_BYTE,0 ;CURSOR POSITION RESTORED |
| FFB7 E804 | 5900 | JMP SHORT EXIT ;INDICATE FINISHED |
| FFB9 5A | 5901 | POP DX ;GET CURSOR POSITION |
| FFBB B402 | 5902 | MOV AH,2 ;TO REQUEST CURSOR SET |
| FFBD C010 | 5903 | INT 10H ;CURSOR POSITION RESTORED |
| FFBF C060000FF | 5904 | MOV STATUS_BYTE,0FFH ;INDICATE ERROR |
| | 5905 | ***** |
| FFC5 5A | 5906 | EXIT: POP DX ;RESTORE ALL THE REGISTERS USED |
| FFC7 59 | 5907 | POP CX |
| FFC9 58 | 5908 | POP BX |
| FFCB 58 | 5909 | POP AX |
| FFCD 1F | 5910 | POP DS |
| FFCF CF | 5911 | IRET |
| | 5912 | ***** |
| FFD1 C017 | 5913 | PRINT_SCREEN ENDP |
| FFD3 C3 | 5914 | ***** |
| | 5915 | 1----- CARRIAGE RETURN, LINE FEED SUBROUTINE |
| | 5916 | ***** |
| FFD5 3302 | 5917 | CRLF PROC NEAR |
| FFD7 32E4 | 5918 | XOR DX,DX ;PRINTER 0 |
| FFD9 B004 | 5919 | XOR AH,AH ;WILL NON SEND INITIAL LF,CR TO PRINTER |
| FFDB C017 | 5920 | MOV AL,120H ;LF |
| FFDD 32E4 | 5921 | INT 17H ;SEND THE LINE FEED |
| FFDF B000 | 5922 | XOR AH,AH ;NON FOR THE CR |
| FFE1 C017 | 5923 | MOV AL,150H ;CR |
| FFE3 C3 | 5924 | INT 17H ;SEND THE CARRIAGE RETURN |
| | 5925 | RET |
| | 5926 | CRLF ENDP |
| | 5927 | CODE ENDS |
| | 5928 | ***** |
| | 5929 | 1----- POWER ON RESET VECTOR |
| | 5930 | ***** |
| FFFF | 5931 | VECTOR SEGMENT AT 0FFFFH |
| | 5932 | ***** |
| | 5933 | 1----- POWER ON RESET |
| | 5934 | ***** |
| 0000 E450000FF | 5935 | JMP RESET |
| 0005 30342F32342F30 | 5936 | DB '04/24/81' ; RELEASE MARKER |
| 31 | 5937 | ***** |
| | 5938 | VECTOR ENDS |
| | 5939 | END |
| | 5940 | ***** |

NOTES

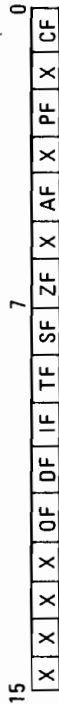
APPENDIX B. ASSEMBLY INSTRUCTION SET REFERENCE

APPENDIX B

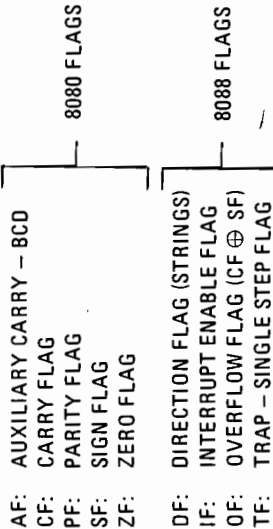
8088 REGISTER MODEL



Instructions which reference the flag register file as a 16-bit object use the symbol FLAGS to represent the file:



X = Don't Care



OPERAND SUMMARY

"reg" field Bit Assignments:

| 16-Bit (w=1) | 8-Bit (w=0) | Segment |
|--------------|-------------|---------|
| 000 AX | 000 AL | 00 ES |
| 001 CX | 001 CL | 01 CS |
| 010 DX | 010 DL | 10 SS |
| 011 BX | 011 BL | 11 DS |
| 100 SP | 100 AH | |
| 101 BP | 101 CH | |
| 110 SI | 110 DH | |
| 111 DI | 111 BH | |

SECOND INSTRUCTION BYTE SUMMARY

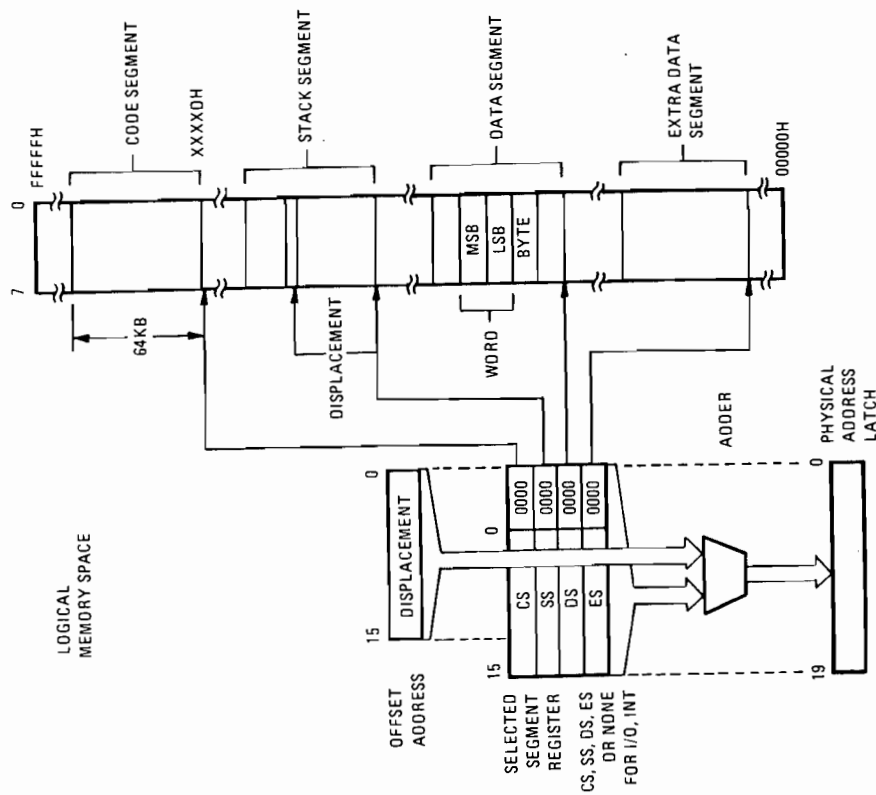
| | | |
|-----|-----|-----|
| mod | xxx | r/m |
|-----|-----|-----|

| mod | Displacement |
|-----|---|
| 00 | DISP = 0*, disp-low and disp-high are absent |
| 01 | DISP = disp-low sign-extended to 16-bits, disp-high is absent |
| 10 | DISP = disp-high: disp-low |
| 11 | r/m is treated as a "reg" field |

| r/m | Operand Address |
|-----|--------------------|
| 000 | (BX) + (SI) + DISP |
| 001 | (BX) + (DI) + DISP |
| 010 | (BP) + (SI) + DISP |
| 011 | (BP) + (DI) + DISP |
| 100 | (SI) + DISP |
| 101 | (DI) + DISP |
| 110 | (BP) + DISP* |
| 111 | (BX) + DISP |

DISP follows 2nd byte of instruction (before data if required).
*except if mod = 00 and r/m = 110 then EA = disp-high: disp-low.

MEMORY SEGMENTATION MODEL



SEGMENT OVERRIDE PREFIX

| | | | | | | |
|---|---|---|-----|---|---|---|
| 0 | 0 | 1 | reg | 1 | 1 | 0 |
|---|---|---|-----|---|---|---|

USE OF SEGMENT OVERRIDE

| OPERAND REGISTER | DEFAULT | WITH OVERRIDE PREFIX |
|---------------------------------------|---------|----------------------|
| IP (code address) | CS | Never |
| SP (stack address) | SS | Never |
| BP (stack address or stack marker) | SS | BP + DS or ES, or CS |
| SI or DI (not incl. strings) | DS | ES, SS, or CS |
| SI (implicit source addr for strings) | DS | ES, SS, or CS |
| DI (implicit dest addr for strings) | ES | Never |

DATA TRANSFER

MOV = Move

Register/memory to/ from register

| | | | | | | | | | |
|---|---|---|---|---|---|---|-----|-----|-----|
| 1 | 0 | 0 | 1 | 0 | d | w | mod | reg | r/m |
|---|---|---|---|---|---|---|-----|-----|-----|

Immediate to register/memory

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|-----|---|---|---|---|-----|------|-------------|
| 1 | 1 | 0 | 0 | 1 | 1 | w | mod | 0 | 0 | 0 | 0 | r/m | data | data if w=1 |
|---|---|---|---|---|---|---|-----|---|---|---|---|-----|------|-------------|

Immediate to register

| | | | | | | | |
|---|---|---|---|---|-----|------|-------------|
| 1 | 0 | 1 | 1 | w | reg | data | data if w=1 |
|---|---|---|---|---|-----|------|-------------|

Memory to accumulator

| | | | | | | | | |
|---|---|---|---|---|---|---|----------|-----------|
| 1 | 0 | 1 | 0 | 0 | 0 | w | addr-low | addr-high |
|---|---|---|---|---|---|---|----------|-----------|

Accumulator to memory

| | | | | | | | | |
|---|---|---|---|---|---|---|----------|-----------|
| 1 | 0 | 1 | 0 | 0 | 1 | w | addr-low | addr-high |
|---|---|---|---|---|---|---|----------|-----------|

Register/memory to segment register

| | | | | | | | | | | |
|---|---|---|---|---|---|---|-----|---|-----|-----|
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | mod | 0 | reg | r/m |
|---|---|---|---|---|---|---|-----|---|-----|-----|

Segment register to register/memory

| | | | | | | | | | | |
|---|---|---|---|---|---|---|-----|---|-----|-----|
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | mod | 0 | reg | r/m |
|---|---|---|---|---|---|---|-----|---|-----|-----|

PUSH = Push

Register/memory

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|-----|---|---|---|-----|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | mod | 1 | 1 | 0 | r/m |
|---|---|---|---|---|---|---|-----|---|---|---|-----|

Register

| | | | | | |
|---|---|---|---|---|-----|
| 0 | 1 | 0 | 1 | 0 | reg |
|---|---|---|---|---|-----|

Segment register

| | | | | | | |
|---|---|---|-----|---|---|---|
| 0 | 0 | 0 | reg | 1 | 1 | 0 |
|---|---|---|-----|---|---|---|

POP = Pop

Register/memory

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|-----|---|---|---|-----|
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | mod | 0 | 0 | 0 | r/m |
|---|---|---|---|---|---|---|-----|---|---|---|-----|

Register

| | | | | | |
|---|---|---|---|---|-----|
| 0 | 1 | 0 | 1 | 1 | reg |
|---|---|---|---|---|-----|

Segment register

| | | | | | | |
|---|---|---|-----|---|---|---|
| 0 | 0 | 0 | reg | 1 | 1 | 1 |
|---|---|---|-----|---|---|---|

XCHG = Exchange

Register/memory with register

| | | | | | | | | | |
|---|---|---|---|---|---|---|-----|-----|-----|
| 1 | 0 | 0 | 0 | 1 | 1 | w | mod | reg | r/m |
|---|---|---|---|---|---|---|-----|-----|-----|

Register with accumulator

| | | | | | |
|---|---|---|---|---|-----|
| 1 | 0 | 0 | 1 | 0 | reg |
|---|---|---|---|---|-----|

IN = Input to AL/AX from

Fixed port

| | | | | | | | | |
|---|---|---|---|---|---|---|---|------|
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | w | port |
|---|---|---|---|---|---|---|---|------|

Variable port (DX)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | w |
|---|---|---|---|---|---|---|---|

OUT = Output from AL/AX to

Fixed port

| | | | | | | | | |
|---|---|---|---|---|---|---|---|------|
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | w | port |
|---|---|---|---|---|---|---|---|------|

Variable port (DX)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | w |
|---|---|---|---|---|---|---|---|

XLAT = Translate byte to AL

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|

LEA = Load EA to register

| | | | | | | | | | |
|---|---|---|---|---|---|---|-----|-----|-----|
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | mod | reg | r/m |
|---|---|---|---|---|---|---|-----|-----|-----|

LDS = Load pointer to DS

| | | | | | | | | | |
|---|---|---|---|---|---|---|-----|-----|-----|
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | mod | reg | r/m |
|---|---|---|---|---|---|---|-----|-----|-----|

LES = Load pointer to ES

| | | | | | | | | | |
|---|---|---|---|---|---|---|-----|-----|-----|
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | mod | reg | r/m |
|---|---|---|---|---|---|---|-----|-----|-----|

LAHF = Load AH with flags

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|

SAHF = Store AH into flags

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
|---|---|---|---|---|---|---|---|

PUSHF = Push flags

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
|---|---|---|---|---|---|---|---|

POPF = Pop flags

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
|---|---|---|---|---|---|---|---|

ARITHMETIC

ADD = Add

Reg./memory with register to either

| | | | | | | | | | |
|---|---|---|---|---|---|---|-----|-----|-----|
| 0 | 0 | 0 | 0 | 0 | d | w | mod | reg | r/m |
|---|---|---|---|---|---|---|-----|-----|-----|

Immediate to register/memory

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|-----|---|---|---|-----|------|----------------|
| 1 | 0 | 0 | 0 | 0 | s | w | mod | 0 | 0 | 0 | r/m | data | data if s:w=01 |
|---|---|---|---|---|---|---|-----|---|---|---|-----|------|----------------|

Immediate to accumulator

| | | | | | | | | |
|---|---|---|---|---|---|---|------|-------------|
| 0 | 0 | 0 | 0 | 1 | 0 | w | data | data if w=1 |
|---|---|---|---|---|---|---|------|-------------|

ADC = Add with carry

Reg./memory with register to either

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|-----|-----|
| 0 | 0 | 0 | 1 | 0 | 0 | d | w | mod | reg | r/m |
|---|---|---|---|---|---|---|---|-----|-----|-----|

Immediate to register/memory

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|-----|---|---|---|-----|------|----------------|
| 1 | 0 | 0 | 0 | 0 | s | w | mod | 0 | 1 | 0 | r/m | data | data if s:w=01 |
|---|---|---|---|---|---|---|-----|---|---|---|-----|------|----------------|

Immediate to accumulator

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|------|-------------|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | w | data | data if w=1 |
|---|---|---|---|---|---|---|---|------|-------------|

INC = Increment

Register/memory

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | w | mod | 0 | 0 | 0 | r/m |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|

Register

| | | | | | |
|---|---|---|---|---|-----|
| 0 | 1 | 0 | 0 | 0 | reg |
|---|---|---|---|---|-----|

AAA = ASCII adjust for add

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|

DAA = Decimal adjust for add

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|

SUB = Subtract

Reg./memory and register to either

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|-----|-----|
| 0 | 0 | 1 | 0 | 1 | 0 | d | w | mod | reg | r/m |
|---|---|---|---|---|---|---|---|-----|-----|-----|

Immediate from register/memory

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|-----|---|---|---|-----|------|----------------|
| 1 | 0 | 0 | 0 | 0 | s | w | mod | 1 | 0 | 1 | r/m | data | data if s:w=01 |
|---|---|---|---|---|---|---|-----|---|---|---|-----|------|----------------|

Immediate from accumulator

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|------|-------------|
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | w | data | data if w=1 |
|---|---|---|---|---|---|---|---|------|-------------|

SBB = Subtract with borrow
Reg./memory and register to either

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|-----|-----|-----|
| 0 | 0 | 0 | 1 | 1 | 0 | d | w | mod | reg | reg | r/m |
|---|---|---|---|---|---|---|---|-----|-----|-----|-----|

Immediate from register/memory

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|-----|---|---|---|-----|------|----------------|
| 1 | 0 | 0 | 0 | 0 | s | w | mod | 0 | 1 | 1 | r/m | data | data if s:w=01 |
|---|---|---|---|---|---|---|-----|---|---|---|-----|------|----------------|

Immediate from accumulator

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|------|-------------|
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | w | data | data if w=1 |
|---|---|---|---|---|---|---|---|------|-------------|

DEC = Decrement

Register/memory

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|-----|---|---|---|-----|
| 1 | 1 | 1 | 1 | 1 | 1 | w | mod | 0 | 0 | 1 | r/m |
|---|---|---|---|---|---|---|-----|---|---|---|-----|

Register

| | | | | | |
|---|---|---|---|---|-----|
| 0 | 1 | 0 | 0 | 1 | reg |
|---|---|---|---|---|-----|

NEG = Change sign

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|-----|---|---|---|-----|
| 1 | 1 | 1 | 0 | 1 | 1 | w | mod | 0 | 1 | 1 | r/m |
|---|---|---|---|---|---|---|-----|---|---|---|-----|

CMP = Compare

Register/memory and register

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|-----|-----|-----|
| 0 | 0 | 1 | 1 | 1 | 0 | d | w | mod | reg | reg | r/m |
|---|---|---|---|---|---|---|---|-----|-----|-----|-----|

Immediate with register/memory

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|-----|---|---|---|-----|------|----------------|
| 1 | 0 | 0 | 0 | 0 | s | w | mod | 1 | 1 | 1 | r/m | data | data if s:w=01 |
|---|---|---|---|---|---|---|-----|---|---|---|-----|------|----------------|

Immediate with accumulator

| | | | | | | | | |
|---|---|---|---|---|---|---|------|-------------|
| 0 | 0 | 1 | 1 | 1 | 0 | w | data | data if w=1 |
|---|---|---|---|---|---|---|------|-------------|

AAS = ASCII adjust for subtract

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|

DAS = Decimal adjust for subtract

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|

MUL = Multiply (unsigned)

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | w | mod | 1 | 0 | 0 | r/m |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|

IMUL = Integer multiply (signed)

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | w | mod | 1 | 0 | 1 | r/m |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|

AAM = ASCII adjust for multiply

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

DIV = Divide (unsigned)

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|-----|---|---|---|-----|
| 1 | 1 | 1 | 0 | 1 | 1 | w | mod | 1 | 1 | 0 | r/m |
|---|---|---|---|---|---|---|-----|---|---|---|-----|

IDIV = Integer divide (signed)

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|-----|---|---|---|-----|
| 1 | 1 | 1 | 0 | 1 | 1 | w | mod | 1 | 1 | 1 | r/m |
|---|---|---|---|---|---|---|-----|---|---|---|-----|

AAD = ASCII adjust for divide

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

CBW = Convert byte to word

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|

CWD = Convert word to double word

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
|---|---|---|---|---|---|---|---|

LOGIC

NOT = Invert

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | w | mod | 0 | 1 | 0 | r/m |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|

SHL/SAL = Shift logical/arithmetic left

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|
| 1 | 1 | 0 | 1 | 0 | 0 | v | w | mod | 1 | 0 | 0 | r/m |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|

SHR = Shift logical right

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|
| 1 | 1 | 0 | 1 | 0 | 0 | v | w | mod | 1 | 0 | 1 | r/m |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|

SAR = Shift arithmetic right

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|
| 1 | 1 | 0 | 1 | 0 | 0 | v | w | mod | 1 | 1 | 1 | r/m |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|

ROL = Rotate left

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|
| 1 | 1 | 0 | 1 | 0 | 0 | v | w | mod | 0 | 0 | 0 | r/m |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|

ROR = Rotate right

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|
| 1 | 1 | 0 | 1 | 0 | 0 | v | w | mod | 0 | 0 | 1 | r/m |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|

RCL = Rotate through carry left

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|
| 1 | 1 | 0 | 1 | 0 | 0 | v | w | mod | 0 | 1 | 0 | r/m |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|

RCR = Rotate through carry right

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|
| 1 | 1 | 0 | 1 | 0 | 0 | v | w | mod | 0 | 1 | 1 | r/m |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|

AND = And

Reg./memory and register to either

| | | | | | | | | | |
|---|---|---|---|---|---|---|-----|-----|-----|
| 0 | 0 | 1 | 0 | 0 | d | w | mod | reg | r/m |
|---|---|---|---|---|---|---|-----|-----|-----|

Immediate to register/memory

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|-----|---|---|---|-----|------|-------------|
| 1 | 0 | 0 | 0 | 0 | 0 | w | mod | 1 | 0 | 0 | r/m | data | data if w=1 |
|---|---|---|---|---|---|---|-----|---|---|---|-----|------|-------------|

Immediate to accumulator

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|------|-------------|
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | w | data | data if w=1 |
|---|---|---|---|---|---|---|---|------|-------------|

TEST = And function to flags, no result

Register/memory and register

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|-----|-----|
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | w | mod | reg | r/m |
|---|---|---|---|---|---|---|---|-----|-----|-----|

Immediate data and register/memory

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|-----|---|---|---|-----|------|-------------|
| 1 | 1 | 1 | 0 | 1 | 1 | w | mod | 0 | 0 | 0 | r/m | data | data if w=1 |
|---|---|---|---|---|---|---|-----|---|---|---|-----|------|-------------|

Immediate data and accumulator

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|------|-------------|
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | w | data | data if w=1 |
|---|---|---|---|---|---|---|---|------|-------------|

OR = Or

Reg./memory and register to either

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|-----|-----|
| 0 | 0 | 0 | 0 | 1 | 0 | d | w | mod | reg | r/m |
|---|---|---|---|---|---|---|---|-----|-----|-----|

Immediate to register/memory

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|-----|---|---|---|-----|------|-------------|
| 1 | 0 | 0 | 0 | 0 | 0 | w | mod | 0 | 0 | 1 | r/m | data | data if w=1 |
|---|---|---|---|---|---|---|-----|---|---|---|-----|------|-------------|

Immediate to accumulator

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|------|-------------|
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | w | data | data if w=1 |
|---|---|---|---|---|---|---|---|------|-------------|

XOR = Exclusive or

Reg./memory and register to either

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|-----|-----|
| 0 | 0 | 1 | 1 | 0 | 0 | d | w | mod | reg | r/m |
|---|---|---|---|---|---|---|---|-----|-----|-----|

Immediate to register/memory

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|-----|---|---|---|-----|------|-------------|
| 1 | 0 | 0 | 0 | 0 | 0 | w | mod | 1 | 1 | 0 | r/m | data | data if w=1 |
|---|---|---|---|---|---|---|-----|---|---|---|-----|------|-------------|

Immediate to accumulator

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|------|-------------|
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | w | data | data if w=1 |
|---|---|---|---|---|---|---|---|------|-------------|

STRING MANIPULATION

REP = Repeat

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | z |
|---|---|---|---|---|---|---|---|

MOVS = Move String

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | w |
|---|---|---|---|---|---|---|---|

CMPS = Compare String

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | w |
|---|---|---|---|---|---|---|---|

SCAS = Scan String

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | w |
|---|---|---|---|---|---|---|---|

LODS = Load String

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | w |
|---|---|---|---|---|---|---|---|

STOS = Store String

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | w |
|---|---|---|---|---|---|---|---|

CONTROL TRANSFER

CALL = Call

Direct within segment

| | | | | | | | | |
|---|---|---|---|---|---|---|----------|-----------|
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | disp-low | disp-high |
|---|---|---|---|---|---|---|----------|-----------|

Indirect within segment

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|-----|---|---|---|-----|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | mod | 0 | 1 | 0 | r/m |
|---|---|---|---|---|---|---|-----|---|---|---|-----|

Direct intersegment

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|------------|-------------|
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | offset-low | offset-high |
| | | | | | | | | seg-low | seg-high |

Indirect intersegment

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|-----|---|---|---|-----|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | mod | 0 | 1 | 1 | r/m |
|---|---|---|---|---|---|---|-----|---|---|---|-----|

JMP = Unconditional Jump

Direct within segment

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|----------|-----------|
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | disp-low | disp-high |
|---|---|---|---|---|---|---|---|----------|-----------|

Direct within segment-short

| | | | | | | | |
|---|---|---|---|---|---|---|------|
| 1 | 1 | 1 | 0 | 1 | 0 | 1 | disp |
|---|---|---|---|---|---|---|------|

Indirect within segment

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|-----|---|---|---|-----|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | mod | 1 | 0 | 0 | r/m |
|---|---|---|---|---|---|---|-----|---|---|---|-----|

Direct intersegment

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|------------|--|-------------|
| 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | offset-low | | offset-high |
| | | | | | | | | seg-low | | seg-high |

Indirect intersegment

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|-----|---|---|---|-----|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | mod | 1 | 0 | 1 | r/m |
|---|---|---|---|---|---|---|-----|---|---|---|-----|

RET = Return from CALL

Within segment

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|

Within seg. adding immed to SP

| | | | | | | | | | |
|---|---|---|---|---|---|---|----------|--|-----------|
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | data-low | | data-high |
|---|---|---|---|---|---|---|----------|--|-----------|

Intersegment

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|

Intersegment, adding immediate to SP

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|----------|--|-----------|
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | data-low | | data-high |
|---|---|---|---|---|---|---|---|----------|--|-----------|

JE/JZ = Jump on equal/zero

| | | | | | | | |
|---|---|---|---|---|---|---|------|
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | disp |
|---|---|---|---|---|---|---|------|

JL/JNGE = Jump on less/not greater or equal

| | | | | | | | |
|---|---|---|---|---|---|---|------|
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | disp |
|---|---|---|---|---|---|---|------|

JLE/JNG = Jump on less or equal/not greater

| | | | | | | | |
|---|---|---|---|---|---|---|------|
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | disp |
|---|---|---|---|---|---|---|------|

JB/JNAE = Jump on below/not above or equal

| | | | | | | | | |
|---|---|---|---|---|---|---|---|------|
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | disp |
|---|---|---|---|---|---|---|---|------|

JBE/JNA = Jump on below or equal/not above

| | | | | | | | | |
|---|---|---|---|---|---|---|---|------|
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | disp |
|---|---|---|---|---|---|---|---|------|

JP/JPE = Jump on parity/parity even

| | | | | | | | |
|---|---|---|---|---|---|---|------|
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | disp |
|---|---|---|---|---|---|---|------|

JO = Jump on overflow

| | | | | | | | |
|---|---|---|---|---|---|---|------|
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | disp |
|---|---|---|---|---|---|---|------|

JS = Jump on sign

| | | | | | | | |
|---|---|---|---|---|---|---|------|
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | disp |
|---|---|---|---|---|---|---|------|

JNE/JNZ = Jump on not equal/not zero

| | | | | | | | |
|---|---|---|---|---|---|---|------|
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | disp |
|---|---|---|---|---|---|---|------|

JNL/JGE = Jump on not less/greater or equal

| | | | | | | | |
|---|---|---|---|---|---|---|------|
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | disp |
|---|---|---|---|---|---|---|------|

JNLE/JG = Jump on not less or equal/greater

| | | | | | | | |
|---|---|---|---|---|---|---|------|
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | disp |
|---|---|---|---|---|---|---|------|

JNB/JAE = Jump on not below/above or equal

| | | | | | | | |
|---|---|---|---|---|---|---|------|
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | disp |
|---|---|---|---|---|---|---|------|

JNBE/JA = Jump on not below or equal/above

| | | | | | | | |
|---|---|---|---|---|---|---|------|
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | disp |
|---|---|---|---|---|---|---|------|

JNP/JPO = Jump on not parity/parity odd

| | | | | | | | |
|---|---|---|---|---|---|---|------|
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | disp |
|---|---|---|---|---|---|---|------|

JNO = Jump on not overflow

| | | | | | | | |
|---|---|---|---|---|---|---|------|
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | disp |
|---|---|---|---|---|---|---|------|

JNS = Jump on not sign

| | | | | | | | |
|---|---|---|---|---|---|---|------|
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | disp |
|---|---|---|---|---|---|---|------|

LOOP = Loop CX times

| | | | | | | | |
|---|---|---|---|---|---|---|------|
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | disp |
|---|---|---|---|---|---|---|------|

LOOPZ/LOOPE = Loop while zero/equal

| | | | | | | | |
|---|---|---|---|---|---|---|------|
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | disp |
|---|---|---|---|---|---|---|------|

LOOPNZ/LOOPNE = Loop while not zero/not equal

| | | | | | | | |
|---|---|---|---|---|---|---|------|
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | disp |
|---|---|---|---|---|---|---|------|

JCXZ = Jump on CX zero

| | | | | | | | |
|---|---|---|---|---|---|---|------|
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | disp |
|---|---|---|---|---|---|---|------|

8088 CONDITIONAL TRANSFER OPERATIONS

| Instruction | Condition | Interpretation |
|-------------|-------------------------|----------------------------------|
| JE or JZ | ZF = 1 | "equal" or "zero" |
| JL or JNGE | (SF xor OF) = 1 | "less" or "not greater or equal" |
| JLE or JNG | ((SF xor OF) or ZF) = 1 | "less or equal" or "not greater" |
| JB or JNAE | CF = 1 | "below" or "not above or equal" |
| JBE or JNA | (CF or ZF) = 1 | "below or equal" or "not above" |
| JP or JPE | PF = 1 | "parity" or "parity even" |
| JO | OF = 1 | "overflow" |
| JS | SF = 1 | "sign" |
| JNE or JNZ | ZF = 0 | "not equal" or "not zero" |
| JNL or JGE | (SF xor OF) = 0 | "not less" or "greater or equal" |
| JNLE or JG | ((SF xor OF) or ZF) = 0 | "not less or equal" or "greater" |
| JNB or JAE | CF = 0 | "not below" or "above or equal" |
| JNBE or JA | (CF or ZF) = 0 | "not below or equal" or "above" |
| JNP or JPO | PF = 0 | "not parity" or "parity odd" |
| JNO | OF = 0 | "not overflow" |
| JNS | SF = 0 | "not sign" |

*"Above" and "below" refer to the relation between two unsigned values, while "greater" and "less" refer to the relation between two signed values.

INT = Interrupt

Type specified

| | | | | | | | | |
|---|---|---|---|---|---|---|---|------|
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | type |
|---|---|---|---|---|---|---|---|------|

Type 3

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
|---|---|---|---|---|---|---|---|

INTO = Interrupt on overflow

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
|---|---|---|---|---|---|---|---|

IRET = Interrupt return

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|

PROCESSOR CONTROL

CLC = Clear carry

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|

STC = Set carry

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
|---|---|---|---|---|---|---|---|---|

CMC = Complement carry

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
|---|---|---|---|---|---|---|---|

NOP = No operation

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|

CLD = Clear direction

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 0 | 0 |
|---|---|---|---|---|---|---|

STD = Set direction

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 0 | 1 |
|---|---|---|---|---|---|---|

CLI = Clear interrupt

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
|---|---|---|---|---|---|---|---|

STI = Set interrupt

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|

HLT = Halt

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
|---|---|---|---|---|---|---|---|

WAIT = Wait

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|

LOCK = Bus lock prefix

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|

ESC = Escape (to external device)

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|
| 1 | 1 | 0 | 1 | 1 | x | x | x | mod | x | x | x | r/m |
|---|---|---|---|---|---|---|---|-----|---|---|---|-----|

Footnotes:

if d = 1 then "to"; if d = 0 then "from"

if w = 1 then word instruction; if w = 0 then byte instruction

if s:w = 01 then 16 bits of immediate data from the operand

if s:w = 11 then an immediate data byte is sign extended to form the 16-bit operand

if v = 0 then "count" = 1; if v = 1 then "count" in (CL)

x = don't care

z is used for some string primitives to compare with ZF FLAG

AL = 8-bit accumulator

AX = 16-bit accumulator

CX = Count register

DS = Data segment

DX = Variable port register

ES = Extra segment

Above/below refers to unsigned value

Greater = more positive;

Less = less positive (more negative) signed values

8088 INSTRUCTION SET MATRIX

| HI | LO | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|----|---------------|-------------|---------|---------|-------------|-------------|-------------|-------------|-------------|
| 0 | ADD | ADD | ADD | ADD | ADD | ADD | ADD | PUSH | POP |
| 1 | ADC | ADC | ADC | ADC | ADC | ADC | ADC | PUSH | POP |
| 2 | AND | AND | AND | AND | AND | AND | AND | SEG | DAA |
| 3 | XOR | XOR | XOR | XOR | XOR | XOR | XOR | SEG | AAA |
| 4 | INC | INC | INC | INC | INC | INC | INC | INC | INC |
| 5 | PUSH | PUSH | PUSH | PUSH | PUSH | PUSH | PUSH | PUSH | PUSH |
| 6 | AX | AX | AX | AX | AX | AX | AX | AX | AX |
| 7 | JO | JNO | JB/JNAE | JNB/JAE | JBE/JNE/JNZ | JBE/JNE/JNA | JBE/JNE/JNA | JBE/JNE/JNA | JBE/JNE/JNA |
| 8 | Immed1 | Immed | Immed | Immed | TEST | TEST | TEST | XCHG | XCHG |
| 9 | NOP | XCHG | XCHG | XCHG | XCHG | XCHG | XCHG | XCHG | XCHG |
| A | MOV | MOV | MOV | MOV | MOV | MOV | MOV | CMPS | CMPS |
| B | MOV | MOV | MOV | MOV | MOV | MOV | MOV | MOV | MOV |
| C | RET | RET | RET | RET | RET | RET | RET | RET | RET |
| D | Shift | Shift | Shift | Shift | Shift | Shift | Shift | Shift | Shift |
| E | LOOPNZ/LOOPNE | LOOPZ/LOOPE | LOOP | LOOP | LOOP | LOOP | LOOP | OUT | OUT |
| F | LOCK | LOCK | LOCK | LOCK | LOCK | LOCK | LOCK | LOCK | LOCK |

b = byte operation
 d = direct
 f = from CPU reg
 i = immediate
 ia = immmed. to accum.
 id = indirect
 is = immmed. byte, sign ext.
 l = long ie. intersegment
 m = memory
 r/m = EA is second byte
 sr = short intrasegment
 t = to CPU reg
 v = variable
 w = word operation
 z = zero

8088 INSTRUCTION SET MATRIX

| HI | LO | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 0 | OR | OR | OR | OR | OR | OR | OR | OR | OR | OR | OR | OR | OR | OR | OR | OR | OR |
| 1 | SBB | SBB | SBB | SBB | SBB | SBB | SBB | SBB | SBB | SBB | SBB | SBB | SBB | SBB | SBB | SBB | SBB |
| 2 | SUB | SUB | SUB | SUB | SUB | SUB | SUB | SUB | SUB | SUB | SUB | SUB | SUB | SUB | SUB | SUB | SUB |
| 3 | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP | CMP |
| 4 | DEC | DEC | DEC | DEC | DEC | DEC | DEC | DEC | DEC | DEC | DEC | DEC | DEC | DEC | DEC | DEC | DEC |
| 5 | POP | POP | POP | POP | POP | POP | POP | POP | POP | POP | POP | POP | POP | POP | POP | POP | POP |
| 6 | AX | AX | AX | AX | AX | AX | AX | AX | AX | AX | AX | AX | AX | AX | AX | AX | AX |
| 7 | JS | JNS | JMP | JMP | JMP | JMP | JMP | JMP | JMP | JMP | JMP | JMP | JMP | JMP | JMP | JMP | JMP |
| 8 | MOV | MOV | MOV | MOV | MOV | MOV | MOV | MOV | MOV | MOV | MOV | MOV | MOV | MOV | MOV | MOV | MOV |
| 9 | CBW | CWD | CALL | CALL | CALL | CALL | CALL | CALL | CALL | CALL | CALL | CALL | CALL | CALL | CALL | CALL | CALL |
| A | TEST | TEST | TEST | TEST | TEST | TEST | TEST | TEST | TEST | TEST | TEST | TEST | TEST | TEST | TEST | TEST | TEST |
| B | MOV | MOV | MOV | MOV | MOV | MOV | MOV | MOV | MOV | MOV | MOV | MOV | MOV | MOV | MOV | MOV | MOV |
| C | RET | RET | RET | RET | RET | RET | RET | RET | RET | RET | RET | RET | RET | RET | RET | RET | RET |
| D | ESC | ESC | ESC | ESC | ESC | ESC | ESC | ESC | ESC | ESC | ESC | ESC | ESC | ESC | ESC | ESC | ESC |
| E | CALL | CALL | CALL | CALL | CALL | CALL | CALL | CALL | CALL | CALL | CALL | CALL | CALL | CALL | CALL | CALL | CALL |
| F | CLC | STC | CLI | CLI | CLI | CLI | CLI | CLI | CLI | CLI | CLI | CLI | CLI | CLI | CLI | CLI | CLI |

where

| mod | r/m | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|-------|-----|------|-----|------|------|---------|-----|------|-----|
| Immed | | ADD | OR | AOC | SBB | AND | SUB | XOR | CMP |
| Shift | | ROL | ROR | RCL | RCR | SHL/SAL | SHR | SAR | |
| Grp 1 | | TEST | NOT | NEG | MUL | IMUL | DIV | IDIV | |
| Grp 2 | | INC | DEC | CALL | CALL | JMP | JMP | PUSH | |

INSTRUCTION SET INDEX

| Mnemonic | Page | Mnemonic | Page | Mnemonic | Page |
|----------|------|----------|------|----------|------|
| AAA | 6 | JG | 12 | MOV | 4 |
| AAD | 8 | JGE | 12 | MOVS | 10 |
| AAM | 8 | JL | 11 | MUL | 7 |
| AAS | 7 | JLE | 11 | NEG | 7 |
| ADC | 6 | JMP | 10 | NOP | 13 |
| ADD | 6 | JNA | 11 | NOT | 8 |
| AND | 9 | JNAE | 11 | OR | 9 |
| CALL | 10 | JNB | 12 | OUT | 5 |
| CBW | 8 | JNBE | 12 | POP | 4 |
| CLC | 13 | JNE | 12 | POPF | 5 |
| CLD | 14 | JNG | 11 | PUSH | 4 |
| CLI | 14 | JNGE | 11 | PUSHF | 5 |
| CMC | 13 | JNL | 12 | RCL | 8 |
| CMP | 7 | JNLE | 12 | RCR | 8 |
| CMPS | 10 | JNO | 12 | REP | 10 |
| CWD | 8 | JNP | 12 | RET | 11 |
| DAA | 6 | JNS | 12 | ROL | 8 |
| DAS | 7 | JNZ | 12 | ROR | 8 |
| DEC | 7 | JO | 11 | SAHF | 5 |
| DIV | 8 | JP | 11 | SAL | 8 |
| ESC | 14 | JPE | 11 | SAR | 8 |
| HLT | 14 | JPO | 12 | SBB | 7 |
| IDIV | 8 | JS | 12 | SCAS | 10 |
| IMUL | 7 | JZ | 11 | SHL | 8 |
| IN | 5 | LAHF | 5 | SHR | 8 |
| INC | 6 | LDS | 5 | STC | 13 |
| INT | 13 | LEA | 5 | STD | 14 |
| INTO | 13 | LES | 5 | STI | 14 |
| IRET | 13 | LOCK | 14 | STOS | 10 |
| JA | 12 | LODS | 10 | SUB | 6 |
| JAE | 12 | LOOP | 12 | TEST | 9 |
| JB | 11 | LOOPE | 12 | WAIT | 14 |
| JBE | 11 | LOOPNE | 12 | XCHG | 5 |
| JCXZ | 12 | LOOPNZ | 12 | XLAT | 5 |
| JE | 11 | LOOPZ | 12 | XOR | 9 |

Appendix C. Of Characters Keystrokes and Color (U.S. Keyboard Layout

| Value | As Characters | | | As Text Attributes | | |
|-------|---------------|--------------|------------------------------------|--------------------------------|----------------------------|--------------------------|
| | | | | Color/Graphics Monitor Adapter | Monochrome Display Adapter | IBM Display Adapter |
| Hex | Dec | Symbol | Keystrokes | Background | Foreground | |
| 00 | 0 | Blank (Null) | Ctrl 2 | Black | Black | Non-Display |
| 01 | 1 | ☺ | Ctrl A | Black | Blue | Underline |
| 02 | 2 | ☹ | Ctrl B | Black | Green | Normal |
| 03 | 3 | ♥ | Ctrl C | Black | Cyan | Normal |
| 04 | 4 | ♦ | Ctrl D | Black | Red | Normal |
| 05 | 5 | ♣ | Ctrl E | Black | Magenta | Normal |
| 06 | 6 | ♠ | Ctrl F | Black | Brown | Normal |
| 07 | 7 | • | Ctrl G | Black | Light Grey | Normal |
| 08 | 8 | • | Ctrl H, Backspace, Shift Backspace | Black | Dark Grey | Non-Display |
| 09 | 9 | ○ | Ctrl I | Black | Light Blue | High Intensity Underline |
| 0A | 10 | ○ | Ctrl J, Ctrl ↓ | Black | Light Green | High Intensity |
| 0B | 11 | ♂ | Ctrl K | Black | Light Green | High Intensity |
| 0C | 12 | ♀ | Ctrl L | Black | Light Red | High Intensity |
| 0D | 13 | ♪ | Ctrl M, ↓, Shift ↓ | Black | Light Magenta | High Intensity |
| 0E | 14 | ♫ | Ctrl N | Black | Yellow | High Intensity |
| 0F | 15 | ☼ | Ctrl O | Black | White | High Intensity |
| 10 | 16 | ▶ | Ctrl P | Blue | Black | Normal |
| 11 | 17 | ◀ | Ctrl Q | Blue | Blue | Underline |
| 12 | 18 | ↕ | Ctrl R | Blue | Green | Normal |
| 13 | 19 | !! | Ctrl S | Blue | Cyan | Normal |
| 14 | 20 | ¶ | Ctrl T | Blue | Red | Normal |
| 15 | 21 | § | Ctrl U | | Magenta | Normal |
| 16 | 22 | ■ | Ctrl V | Blue | Brown | Normal |
| 17 | 23 | ↕ | Ctrl W | Blue | Light Grey | Normal |

| Value | | As Characters | | | | As Text Attributes | | |
|-------|-----|--------------------------------|--|--------|--------------------------------|--------------------|--------------------------|--|
| | | Color/Graphics Monitor Adapter | | | IBM Monochrome Display Adapter | | | |
| Hex | Dec | Symbol | Keystrokes | Modes | Background | Foreground | | |
| 18 | 24 | ! | Ctrl X | | Blue | Dark Grey | High Intensity | |
| 19 | 25 | , | Ctrl Y | | Blue | Light Blue | High Intensity Underline | |
| 1A | 26 | - | Ctrl Z | | Blue | Light Green | High Intensity | |
| 1B | 27 | - | Ctrl [, Esc, Shift Esc, Ctrl Esc | | Blue | Light Cyan | High Intensity | |
| 1C | 28 | _ | Ctrl \ | | Blue | Light Red | High Intensity | |
| 1D | 29 | ↔ | Ctrl } | | Blue | Light Magenta | High Intensity | |
| 1E | 30 | ▲ | Ctrl 6 | | Blue | Yellow | High Intensity | |
| 1F | 31 | ▼ | Ctrl - | | Blue | White | High Intensity | |
| 20 | 32 | Blank Space | Space Bar, Shift, Space, Ctrl Space, Alt Space | | Green | Black | Normal | |
| 21 | 33 | | | Shift | Green | Blue | Underline | |
| 22 | 34 | " | " | Shift | Green | Green | Normal | |
| 23 | 35 | # | # | Shift | Green | Cyan | Normal | |
| 24 | 36 | \$ | \$ | Shift | Green | Red | Normal | |
| 25 | 37 | % | % | Shift | Green | Magenta | Normal | |
| 26 | 38 | & | & | Shift | Green | Brown | Normal | |
| 27 | 39 | . | . | | Green | Light Grey | Normal | |
| 28 | 40 | (| (| Shift | Green | Dark Grey | High Intensity | |
| 29 | 41 |) |) | Shift | Green | Light Blue | High Intensity Underline | |
| 2A | 42 | * | * | Note 1 | Green | Light Green | High Intensity | |
| 2B | 43 | + | + | Shift | Green | Light Cyan | High Intensity | |
| 2C | 44 | . | . | | Green | Light Red | High Intensity | |
| 2D | 45 | — | — | | Green | Light Magenta | High Intensity | |
| 2E | 46 | . | . | Note 2 | Green | Yellow | High Intensity | |

























| Value | | As Characters | | | | As Text Attributes | | |
|-------|-----|---------------|------------|--------|--------------------------------|--------------------|--------------------------------|--|
| | | Symbol | Keystrokes | Modes | Color/Graphics Monitor Adapter | | IBM Monochrome Display Adapter | |
| Hex | Dec | | | | Background | Foreground | | |
| 2F | 47 | / | / | | Green | White | High Intensity | |
| 30 | 48 | 0 | 0 | Note 3 | Cyan | Black | Normal | |
| 31 | 49 | 1 | 1 | Note 3 | Cyan | Blue | Underline | |
| 32 | 50 | 2 | 2 | Note 3 | Cyan | Green | Normal | |
| 33 | 51 | 3 | 3 | Note 3 | Cyan | Cyan | Normal | |
| 34 | 52 | 4 | 4 | Note 3 | Cyan | Red | Normal | |
| 35 | 53 | 5 | 5 | Note 3 | Cyan | Magenta | Normal | |
| 36 | 54 | 6 | 6 | Note 3 | Cyan | Brown | Normal | |
| 37 | 55 | 7 | 7 | Note 3 | Cyan | Light Grey | Normal | |
| 38 | 56 | 8 | 8 | Note 3 | Cyan | Dark Grey | High Intensity | |
| 39 | 57 | 9 | 9 | Note 3 | Cyan | Light Blue | High Intensity Underline | |
| 3A | 58 | : | : | Shift | Cyan | Light Green | High Intensity | |
| 3B | 59 | ; | ; | | Cyan | Light Cyan | High Intensity | |
| 3C | 60 | < | < | Shift | Cyan | Light Red | High Intensity | |
| 3D | 61 | = | = | | Cyan | Light Magenta | High Intensity | |
| 3E | 62 | > | > | Shift | Cyan | Yellow | High Intensity | |
| 3F | 63 | ? | ? | Shift | Cyan | White | High Intensity | |
| 40 | 64 | @ | @ | Shift | Red | Black | Normal | |
| 41 | 65 | A | A | Note 4 | Red | Blue | Underline | |
| 42 | 66 | B | B | Note 4 | Red | Green | Normal | |
| 43 | 67 | C | C | Note 4 | Red | Cyan | Normal | |
| 44 | 68 | D | D | Note 4 | Red | Red | Normal | |
| 45 | 69 | E | E | Note 4 | Red | Magenta | Normal | |
| 46 | 70 | F | F | Note 4 | Red | Brown | Normal | |
| 47 | 71 | G | G | Note 4 | Red | Light Grey | Normal | |
| 48 | 72 | H | H | Note 4 | Red | Dark Grey | High Intensity | |
| 49 | 73 | I | I | Note 4 | Red | Light Blue | High Intensity Underline | |
| 4A | 74 | J | J | Note 4 | Red | Light Green | High Intensity | |

| Value | | As Characters | | | | As Text Attributes | |
|-------|-----|--------------------------------|------------|--------------------------------|------------|--------------------|--|
| | | Color/Graphics Monitor Adapter | | IBM Monochrome Display Adapter | | | |
| Hex | Dec | Symbol | Keystrokes | Modes | Background | Foreground | |
| 4B | 75 | K | K | Note 4 | Red | Light Cyan | |
| 4C | 76 | L | L | Note 4 | Red | Light Red | |
| 4D | 77 | M | M | Note 4 | Red | Light Magenta | |
| 4E | 78 | N | N | Note 4 | Red | Yellow | |
| 4F | 79 | O | O | Note 4 | Red | White | |
| 50 | 80 | P | P | Note 4 | Magenta | Black | |
| 51 | 81 | Q | Q | Note 4 | Magenta | Blue | |
| 52 | 82 | R | R | Note 4 | Magenta | Green | |
| 53 | 83 | S | S | Note 4 | Magenta | Cyan | |
| 54 | 84 | T | T | Note 4 | Magenta | Red | |
| 55 | 85 | U | U | Note 4 | Magenta | Magenta | |
| 56 | 86 | V | V | Note 4 | Magenta | Brown | |
| 57 | 87 | W | W | Note 4 | Magenta | Light Grey | |
| 58 | 88 | X | X | Note 4 | Magenta | Dark Grey | |
| 59 | 89 | Y | Y | Note 4 | Magenta | Light Blue | |
| 5A | 90 | Z | Z | Note 4 | Magenta | Light Green | |
| 5B | 91 | [| [| | Magenta | Light Cyan | |
| 5C | 92 | \ | \ | | Magenta | Light Red | |
| 5D | 93 |] |] | | Magenta | Light Magenta | |
| 5E | 94 | ^ | ^ | Shift | Magenta | Yellow | |
| 5F | 95 | — | — | Shift | Magenta | White | |
| 60 | 96 | · | · | | Yellow | Black | |
| 61 | 97 | a | a | Note 5 | Yellow | Blue | |
| 62 | 98 | b | b | Note 5 | Yellow | Green | |
| 63 | 99 | c | c | Note 5 | Yellow | Cyan | |
| 64 | 100 | d | d | Note 5 | Yellow | Red | |
| 65 | 101 | e | e | Note 5 | Yellow | Magenta | |
| 66 | 102 | f | f | Note 5 | Yellow | Brown | |

| Value | As Characters | | | | As Text Attributes | | |
|-------|---------------|-----|--------|------------|--------------------|------------|-----------------------------------|
| | Hex | Dec | Symbol | Keystrokes | Modes | Background | Color/Graphics Monitor Adapter |
| 67 | 103 | g | g | g | Note 5 | Yellow | Light Grey |
| 68 | 104 | h | h | h | Note 5 | Yellow | Dark Grey |
| 69 | 105 | i | i | i | Note 5 | Yellow | Light Blue |
| 6A | 106 | j | j | j | Note 5 | Yellow | Light Green |
| 6B | 107 | k | k | k | Note 5 | Yellow | Light Cyan |
| 6C | 108 | l | l | l | Note 5 | Yellow | Light Red |
| 6D | 109 | m | m | m | Note 5 | Yellow | Light Magenta |
| 6E | 110 | n | n | n | Note 5 | Yellow | Yellow |
| 6F | 111 | o | o | o | Note 5 | Yellow | White |
| 70 | 112 | p | p | p | Note 5 | White | Black |
| 71 | 113 | q | q | q | Note 5 | White | Blue |
| 72 | 114 | r | r | r | Note 5 | White | Green |
| 73 | 115 | s | s | s | Note 5 | White | Cyan |
| 74 | 116 | f | f | f | Note 5 | White | Red |
| 75 | 117 | u | u | u | Note 5 | White | Magenta |
| 76 | 118 | v | v | v | Note 5 | White | Brown |
| 77 | 119 | w | w | w | Note 5 | White | Light Grey |
| 78 | 120 | x | x | x | Note 5 | White | Dark Grey |
| 79 | 121 | y | y | y | Note 5 | White | Light Blue |
| 7A | 122 | z | z | z | Note 5 | White | Light Green |
| 7B | 123 | { | { | { | Shift | White | Light Cyan |
| 7C | 124 | | | | Shift | White | Light Red |
| 7D | 125 | } | } | } | Shift | White | Light Magenta |
| 7E | 126 | ~ | ~ | ~ | Shift | White | Yellow |
| 7F | 127 | Δ | Δ | Ctrl — | | White | White |

| Value | As Characters | | | | As Text Attributes | | |
|--------|---------------|-----|--------|------------|--------------------|--------------------------------|--------------------------------|
| | Hex | Dec | Symbol | Keystrokes | Modes | Color/Graphics Monitor Adapter | IBM Monochrome Display Adapter |
| 9B 155 | | | ¢ | Alt 155 | Note 6 | Blue | Light Cyan |
| 9C 156 | | | £ | Alt 156 | Note 6 | Blue | Light Red |
| 9D 157 | | | ¥ | Alt 157 | Note 6 | Blue | Light Magenta |
| 9E 158 | | | ₤ | Alt 158 | Note 6 | Blue | Yellow |
| 9F 159 | | | ₥ | Alt 159 | Note 6 | Blue | White |
| A0 160 | | | ₦ | Alt 160 | Note 6 | Green | Black |
| A1 161 | | | ₯ | Alt 161 | Note 6 | Green | Blue |
| A2 162 | | | ₰ | Alt 162 | Note 6 | Green | Green |
| A3 163 | | | ₱ | Alt 163 | Note 6 | Green | Cyan |
| A4 164 | | | ₲ | Alt 164 | Note 6 | Green | Red |
| A5 165 | | | ₳ | Alt 165 | Note 6 | Green | Magenta |
| A6 166 | | | ₴ | Alt 166 | Note 6 | Green | Brown |
| A7 167 | | | ₵ | Alt 167 | Note 6 | Green | Light Grey |
| A8 168 | | | ₶ | Alt 168 | Note 6 | Green | Dark Grey |
| A9 169 | | | ₷ | Alt 169 | Note 6 | Green | Light Blue |
| AA 170 | | | ₸ | Alt 170 | Note 6 | Green | Light Green |
| AB 171 | | | ₹ | Alt 171 | Note 6 | Green | Light Cyan |
| AC 172 | | | ₺ | Alt 172 | Note 6 | Green | Light Red |
| AD 173 | | | ₻ | Alt 173 | Note 6 | Green | Light Magenta |
| AE 174 | | | ₼ | Alt 174 | Note 6 | Green | Yellow |
| AF 175 | | | ₽ | Alt 175 | Note 6 | Green | White |
| B0 176 | | | ₾ | Alt 176 | Note 6 | Cyan | Black |
| B1 177 | | | ₿ | Alt 177 | Note 6 | Cyan | Blue |
| B2 178 | | | ₰ | Alt 178 | Note 6 | Cyan | Green |
| B3 179 | | | ₱ | Alt 179 | Note 6 | Cyan | Cyan |
| B4 180 | | | ₲ | Alt 180 | Note 6 | Cyan | Red |
| B5 181 | | | ₳ | Alt 181 | Note 6 | Cyan | Magenta |
| B6 182 | | | ₴ | Alt 182 | Note 6 | Cyan | Brown |

| Value | As Characters | | | | As Text Attributes | | |
|--------|---------------|-----|--------|------------|--------------------|--------------------------------|--------------------------------|
| | Hex | Dec | Symbol | Keystrokes | Modes | Color/Graphics Monitor Adapter | IBM Monochrome Display Adapter |
| 80 128 | | | Ç | Alt 128 | Note 6 | Black | Non-Display |
| 81 129 | | | ü | Alt 129 | Note 6 | Black | Underline |
| 82 130 | | | é | Alt 130 | Note 6 | Black | Normal |
| 83 131 | | | à | Alt 131 | Note 6 | Black | Normal |
| 84 132 | | | á | Alt 132 | Note 6 | Black | Normal |
| 85 133 | | | â | Alt 133 | Note 6 | Black | Normal |
| 86 134 | | | ã | Alt 134 | Note 6 | Black | Normal |
| 87 135 | | | ç | Alt 135 | Note 6 | Black | Normal |
| 88 136 | | | ê | Alt 136 | Note 6 | Black | Non-Display |
| 89 137 | | | ë | Alt 137 | Note 6 | Black | High Intensity Underline |
| 8A 138 | | | è | Alt 138 | Note 6 | Black | High Intensity |
| 8B 139 | | | í | Alt 139 | Note 6 | Black | High Intensity |
| 8C 140 | | | ì | Alt 140 | Note 6 | Black | High Intensity |
| 8D 141 | | | ï | Alt 141 | Note 6 | Black | High Intensity |
| 8E 142 | | | À | Alt 142 | Note 6 | Black | High Intensity |
| 8F 143 | | | Á | Alt 143 | Note 6 | Black | High Intensity |
| 90 144 | | | Ê | Alt 144 | Note 6 | Blue | Normal |
| 91 145 | | | æ | Alt 145 | Note 6 | Blue | Underline |
| 92 146 | | | Æ | Alt 146 | Note 6 | Blue | Normal |
| 93 147 | | | Ö | Alt 147 | Note 6 | Blue | Normal |
| 94 148 | | | ö | Alt 148 | Note 6 | Blue | Normal |
| 95 149 | | | ó | Alt 149 | Note 6 | Blue | Normal |
| 96 150 | | | û | Alt 150 | Note 6 | Blue | Normal |
| 97 151 | | | ü | Alt 151 | Note 6 | Blue | Normal |
| 98 152 | | | ÿ | Alt 152 | Note 6 | Blue | High Intensity |
| 99 153 | | | ø | Alt 153 | Note 6 | Blue | High Intensity Underline |
| 9A 154 | | | ù | Alt 154 | Note 6 | Blue | High Intensity |

| Value | | As Characters | | | | | As Text Attributes | | |
|-------|-----|---------------|---|---------|------------|---------|--------------------------------|--------------------------|--------------------------------|
| | | Hex | Dec | Symbol | Keystrokes | Modes | Color/Graphics Monitor Adapter | | IBM Monochrome Display Adapter |
| B7 | 183 | |  | Alt 183 | Note 6 | Cyan | Light Grey | Normal | |
| B8 | 184 | |  | Alt 184 | Note 6 | Cyan | Dark Grey | High Intensity | |
| B9 | 185 | |  | Alt 185 | Note 6 | Cyan | Light Blue | High Intensity Underline | |
| BA | 186 | |  | Alt 186 | Note 6 | Cyan | Light Green | High Intensity | |
| BB | 187 | |  | Alt 187 | Note 6 | Cyan | Light Cyan | High Intensity | |
| BC | 188 | |  | Alt 188 | Note 6 | Cyan | Light Red | High Intensity | |
| BD | 189 | |  | Alt 189 | Note 6 | Cyan | Light Magenta | High Intensity | |
| BE | 190 | |  | Alt 190 | Note 6 | Cyan | Yellow | High Intensity | |
| BF | 191 | |  | Alt 191 | Note 6 | Cyan | White | High Intensity | |
| C0 | 192 | |  | Alt 192 | Note 6 | Red | Black | Normal | |
| C1 | 193 | |  | Alt 193 | Note 6 | Red | Blue | Underline | |
| C2 | 194 | |  | Alt 194 | Note 6 | Red | Green | Normal | |
| C3 | 195 | |  | Alt 195 | Note 6 | Red | Cyan | Normal | |
| C4 | 196 | |  | Alt 196 | Note 6 | Red | Red | Normal | |
| C5 | 197 | |  | Alt 197 | Note 6 | Red | Magenta | Normal | |
| C6 | 198 | |  | Alt 198 | Note 6 | Red | Brown | Normal | |
| C7 | 199 | |  | Alt 199 | Note 6 | Red | Light Grey | Normal | |
| C8 | 200 | |  | Alt 200 | Note 6 | Red | Dark Grey | High Intensity | |
| C9 | 201 | |  | Alt 201 | Note 6 | Red | Light Blue | High Intensity Underline | |
| CA | 202 | |  | Alt 202 | Note 6 | Red | Light Green | High Intensity | |
| CB | 203 | |  | Alt 203 | Note 6 | Red | Light Cyan | High Intensity | |
| CC | 204 | |  | Alt 204 | Note 6 | Red | Light Red | High Intensity | |
| CD | 205 | |  | Alt 205 | Note 6 | Red | Light Magenta | High Intensity | |
| CE | 206 | |  | Alt 206 | Note 6 | Red | Yellow | High Intensity | |
| CF | 207 | | | Alt 207 | Note 6 | Red | White | High Intensity | |
| D0 | 208 | | | Alt 208 | Note 6 | Magenta | Black | Normal | |

| Value | | As Characters | | | | As Text Attributes | | | |
|-------|-----|---------------|-----|---------|------------|--------------------|------------|---------------|--------------------------------|
| | | Hex | Dec | Symbol | Keystrokes | Modes | Background | Foreground | IBM Monochrome Display Adapter |
| D1 | 209 | | | Alt 209 | Note 6 | | Magenta | Blue | Underline |
| D2 | 210 | | | Alt 210 | Note 6 | | Magenta | Green | Normal |
| D3 | 211 | | | Alt 211 | Note 6 | | Magenta | Cyan | Normal |
| D4 | 212 | | | Alt 212 | Note 6 | | Magenta | Red | Normal |
| D5 | 213 | | | Alt 213 | Note 6 | | Magenta | Magenta | Normal |
| D6 | 214 | | | Alt 214 | Note 6 | | Magenta | Brown | Normal |
| D7 | 215 | | | Alt 215 | Note 6 | | Magenta | Light Grey | Normal |
| D8 | 216 | | | Alt 216 | Note 6 | | Magenta | Dark Grey | High Intensity |
| D9 | 217 | | | Alt 217 | Note 6 | | Magenta | Light Blue | High Intensity Underline |
| DA | 218 | | | Alt 218 | Note 6 | | Magenta | Light Green | High Intensity |
| DB | 219 | | | Alt 219 | Note 6 | | Magenta | Light Cyan | High Intensity |
| DC | 220 | | | Alt 220 | Note 6 | | Magenta | Light Red | High Intensity |
| DD | 221 | | | Alt 221 | Note 6 | | Magenta | Light Magenta | High Intensity |
| DE | 222 | | | Alt 222 | Note 6 | | Magenta | Yellow | High Intensity |
| DF | 223 | | | Alt 223 | Note 6 | | Magenta | White | High Intensity |
| E0 | 224 | | | Alt 224 | Note 6 | | Yellow | Black | Normal |
| E1 | 225 | | | Alt 225 | Note 6 | | Yellow | Blue | Underline |
| E2 | 226 | | | Alt 226 | Note 6 | | Yellow | Green | Normal |
| E3 | 227 | | | Alt 227 | Note 6 | | Yellow | Cyan | Normal |
| E4 | 228 | | | Alt 228 | Note 6 | | Yellow | Red | Normal |
| E5 | 229 | | | Alt 229 | Note 6 | | Yellow | Magenta | Normal |
| E6 | 230 | | | Alt 230 | Note 6 | | Yellow | Brown | Normal |
| E7 | 231 | | | Alt 231 | Note 6 | | Yellow | Light Grey | Normal |
| E8 | 232 | | | Alt 232 | Note 6 | | Yellow | Dark Grey | High Intensity |
| E9 | 233 | | | Alt 233 | Note 6 | | Yellow | Light Blue | High Intensity Underline |
| EA | 234 | | | Alt 234 | Note 6 | | Yellow | Light Green | High Intensity |
| EB | 235 | | | Alt 235 | Note 6 | | Yellow | Light Cyan | High Intensity |

| Value | As Characters | | | | | As Text Attributes | | | |
|-------|---------------|-------|---------|------------|--------|--------------------------------|---------------|--------------------------------|--|
| | Hex | Dec | Symbol | Keystrokes | Modes | Color/Graphics Monitor Adapter | | IBM Monochrome Display Adapter | |
| EC | 236 | ∞ | Alt 236 | Note 6 | Yellow | Background | Foreground | High Intensity | |
| ED | 237 | ϕ | Alt 237 | Note 6 | Yellow | Yellow | Light Magenta | High Intensity | |
| EE | 238 | € | Alt 238 | Note 6 | Yellow | Yellow | Yellow | High Intensity | |
| EF | 239 | ∩ | Alt 239 | Note 6 | Yellow | White | White | High Intensity | |
| FO | 240 | ≡ | Alt 240 | Note 6 | White | White | Black | Reverse Video | |
| F1 | 241 | ± | Alt 241 | Note 6 | White | White | Blue | Underline | |
| F2 | 242 | ≈ | Alt 242 | Note 6 | White | White | Green | Normal | |
| F3 | 243 | ≤ | Alt 243 | Note 6 | White | White | Cyan | Normal | |
| F4 | 244 | ∫ | Alt 244 | Note 6 | White | White | Red | Normal | |
| F5 | 245 | ∫ | Alt 245 | Note 6 | White | White | Magenta | Normal | |
| F6 | 246 | ÷ | Alt 246 | Note 6 | White | White | Brown | Normal | |
| F7 | 247 | ≈ | Alt 247 | Note 6 | White | White | Light Grey | Normal | |
| F8 | 248 | ○ | Alt 248 | Note 6 | White | White | Dark Grey | Reverse Video | |
| F9 | 249 | ● | Alt 249 | Note 6 | White | White | Light Blue | High Intensity Underline | |
| FA | 250 | • | Alt 250 | Note 6 | White | White | Light Green | High Intensity | |
| FB | 251 | √ | Alt 251 | Note 6 | White | White | Light Cyan | High Intensity | |
| FC | 252 | η | Alt 252 | Note 6 | White | White | Light Red | High Intensity | |
| FD | 253 | 2 | Alt 253 | Note 6 | White | White | Light Magenta | High Intensity | |
| FE | 254 | ■ | Alt 254 | Note 6 | White | White | Yellow | High Intensity | |
| FF | 255 | BLANK | Alt 255 | Note 6 | White | White | White | High Intensity | |

NOTE 1 Asterisk (*) can easily be keyed using two methods:
1) hit the **Prt Sc** key or 2) in shift mode hit the **8** key.

NOTE 2 Period (.) can easily be keyed using two methods:
1) hit the **>** key or 2) in shift or Num Lock mode hit the **Del** key.

NOTE 3 Numeric characters (0—9) can easily be keyed using two methods: 1) hit the numeric keys on the top row of the typewriter portion of the keyboard or 2) in shift or Num Lock mode hit the numeric keys in the 10—key pad portion of the keyboard.

NOTE 4 Upper case alphabetic characters (A—Z) can easily be keyed in two modes: 1) in shift mode the appropriate alphabetic key or 2) in Caps Lock mode hit the appropriate alphabetic key.

NOTE 5 Lower case alphabetic characters (a—z) can easily be keyed in two modes: 1) in "normal" mode hit the appropriate key or 2) in Caps Lock combined with shift mode hit the appropriate alphabetic key.

NOTE 6 The 3 digits after the Alt key must be typed from the numeric key pad (keys 71—73, 75—77, 79—82). Character codes 000 through 255 can be entered in this fashion. (With Caps Lock activated, Character codes 97 through 122 will display upper case rather than lower case alphabetic characters.)

Character Set (00-7F) Quick Reference

| DECIMAL VALUE | HEXA DECIMAL VALUE | 0 | 16 | 32 | 48 | 64 | 80 | 96 | 112 |
|---------------|--------------------|--------------|----|---------------|----|----|----|----|-----|
| 0 | 0 | BLANK (NULL) | ► | BLANK (SPACE) | 0 | @ | P | , | p |
| 1 | 1 | ☺ | ◄ | ! | 1 | A | Q | a | q |
| 2 | 2 | ☹ | ↕ | " | 2 | B | R | b | r |
| 3 | 3 | ♥ | !! | # | 3 | C | S | c | s |
| 4 | 4 | ♦ | ⌵ | \$ | 4 | D | T | d | t |
| 5 | 5 | ♣ | § | % | 5 | E | U | e | u |
| 6 | 6 | ♠ | ■ | & | 6 | F | V | f | v |
| 7 | 7 | • | ↑↓ | ' | 7 | G | W | w | w |
| 8 | 8 | • | ↑ | (| 8 | H | X | h | x |
| 9 | 9 | ○ | ↓ |) | 9 | I | Y | i | y |
| 10 | A | ○ | → | * | : | J | Z | j | z |
| 11 | B | ♂ | ← | + | ; | K | I | k | { |
| 12 | C | ♀ | └ | , | < | L | / | l | ! |
| 13 | D | ♪ | ↔ | — | = | M | I | m | } |
| 14 | E | ♪ | ◄ | . | > | N | ^ | n | ~ |
| 15 | F | ☼ | ◄ | / | ? | O | _ | o | Δ |

Character Set (80-FF) Quick Reference

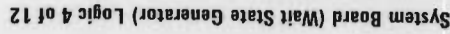
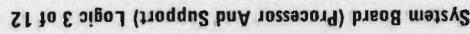
| DECIMAL VALUE | HEXA DECIMAL VALUE | 128 | 144 | 160 | 176 | 192 | 208 | 224 | 240 |
|---------------|--------------------|-----|-----|-----|-----|-----|-----|-----|------------|
| 0 | 0 | Ç | É | á | ▤ | ▤ | ▤ | ∞ | F |
| 1 | 1 | ü | æ | í | ▥ | ▥ | ▥ | β | |
| 2 | 2 | é | Æ | ó | ▦ | ▦ | ▦ | Γ | |
| 3 | 3 | â | ô | ú | ▧ | ▧ | ▧ | π | |
| 4 | 4 | ä | ö | ñ | ▨ | ▨ | ▨ | Σ | |
| 5 | 5 | á | ò | Ñ | ▩ | ▩ | ▩ | σ | |
| 6 | 6 | ã | û | ä | ▪ | ▪ | ▪ | ρ | |
| 7 | 7 | ç | ù | ó | ▫ | ▫ | ▫ | τ | |
| 8 | 8 | ê | ÿ | ï | ▬ | ▬ | ▬ | ø | |
| 9 | 9 | ë | Ö | ┐ | ▭ | ▭ | ▭ | θ | |
| 10 | A | è | Ü | ┐ | ▮ | ▮ | ▮ | Ω | |
| 11 | B | ï | ç | ½ | ▯ | ▯ | ▯ | δ | |
| 12 | C | î | £ | ¼ | ▰ | ▰ | ▰ | ∞ | |
| 13 | D | ì | ¥ | ì | ▱ | ▱ | ▱ | φ | |
| 14 | E | Ä | ℞ | « | ▲ | ▲ | ▲ | ∈ | |
| 15 | F | Å | f | » | △ | △ | △ | ∪ | BLANK 'FF' |

APPENDIX D. LOGIC DIAGRAMS

Contents

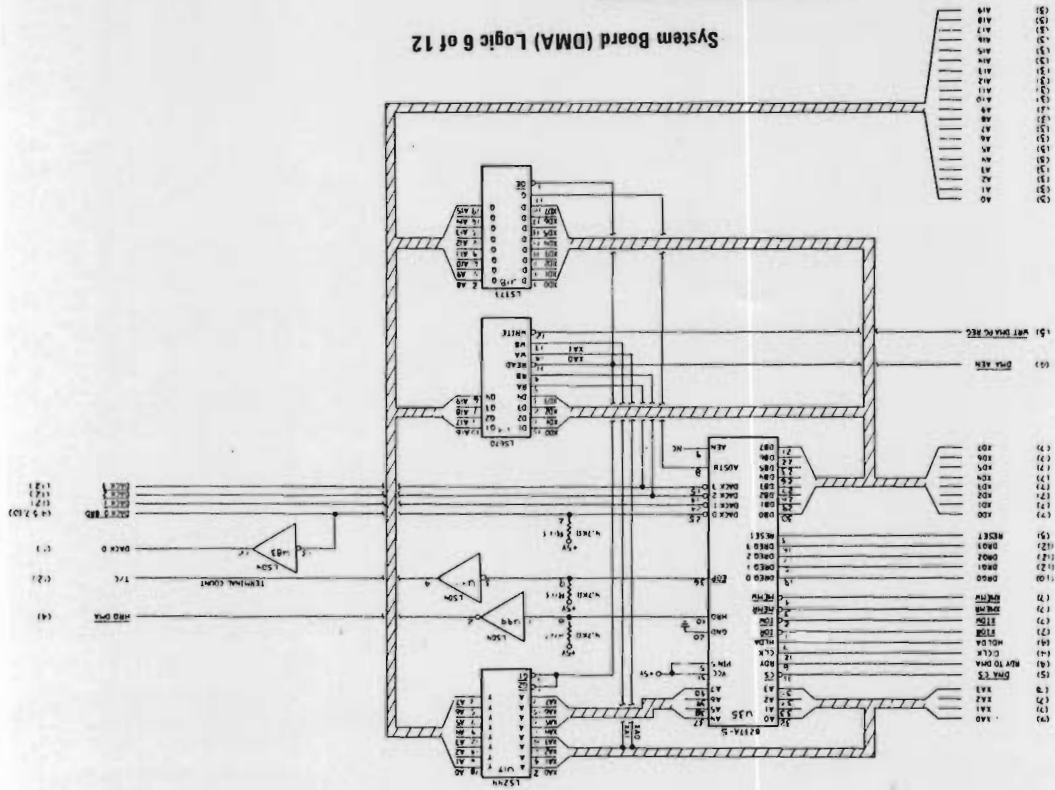
| | |
|---|------|
| System Board | D-2 |
| Keyboard | D-12 |
| IBM Monochrome Display And Parallel Printer Adapter | D-14 |
| IBM Monochrome Display | D-24 |
| Color/Graphics Monitor Adapter | D-27 |
| IBM 80 CPS Graphics Printer | D-33 |
| Parallel Printer Adapter | D-36 |
| 5 1/4" Diskette Drive Adapter | D-37 |
| 5 1/4" Diskette Drive | D-41 |
| 32KB Memory Expansion Option | D-44 |
| 64KB Memory Expansion Option | D-47 |
| 64/256KB Memory Expansion Option | D-50 |
| Asynchronous Communications Adapter | D-54 |
| Game Control Adapter | D-55 |
| Prototype Card | D-56 |

SYSTEM BOARD (PROCESSOR AND SUPPORT)

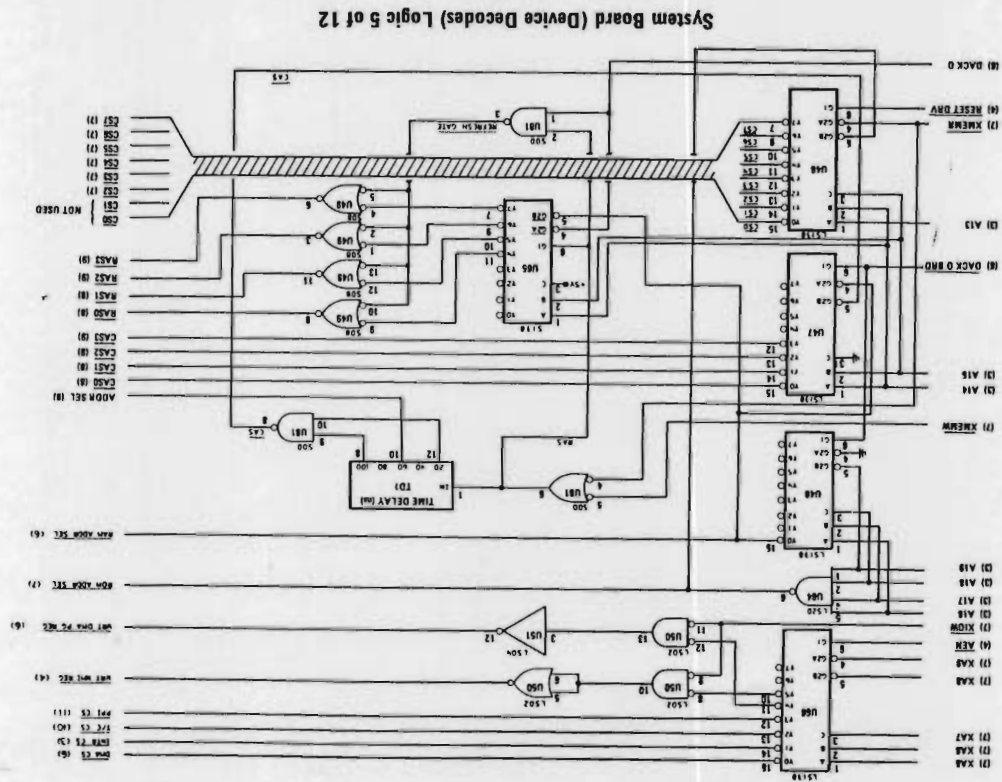


D-2

SYSTEM BOARD (DMA)

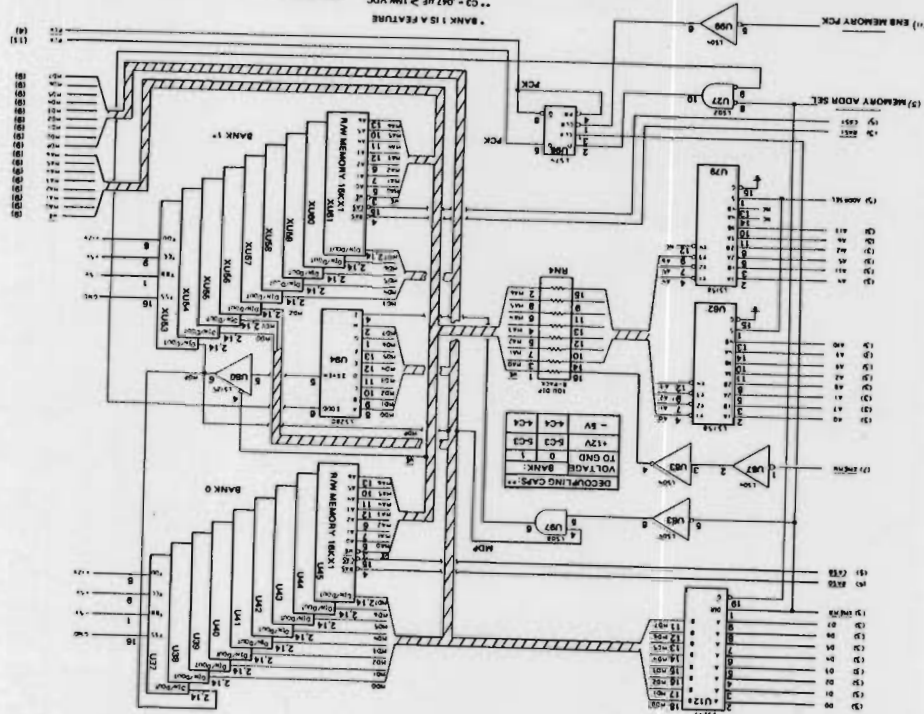


SYSTEM BOARD (DEVICE DECODES)



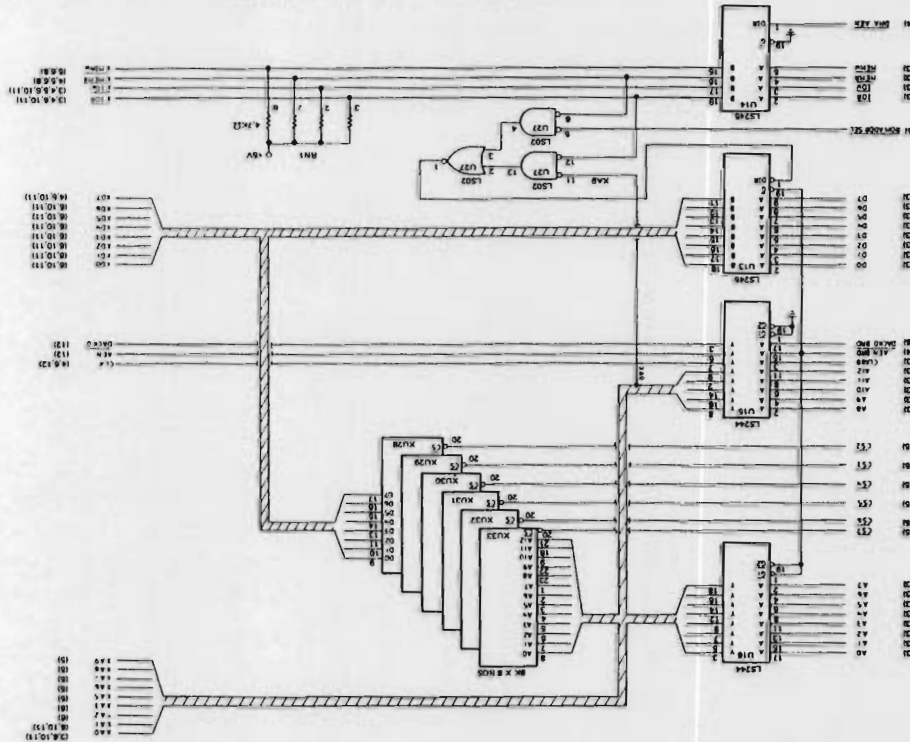
System Board (Dynamic Memory) Logic 8 of 12

** BANK 1 IS A FEATURE
C3 = 0.047 μ F \geq 10W VDC
DECOUPLING



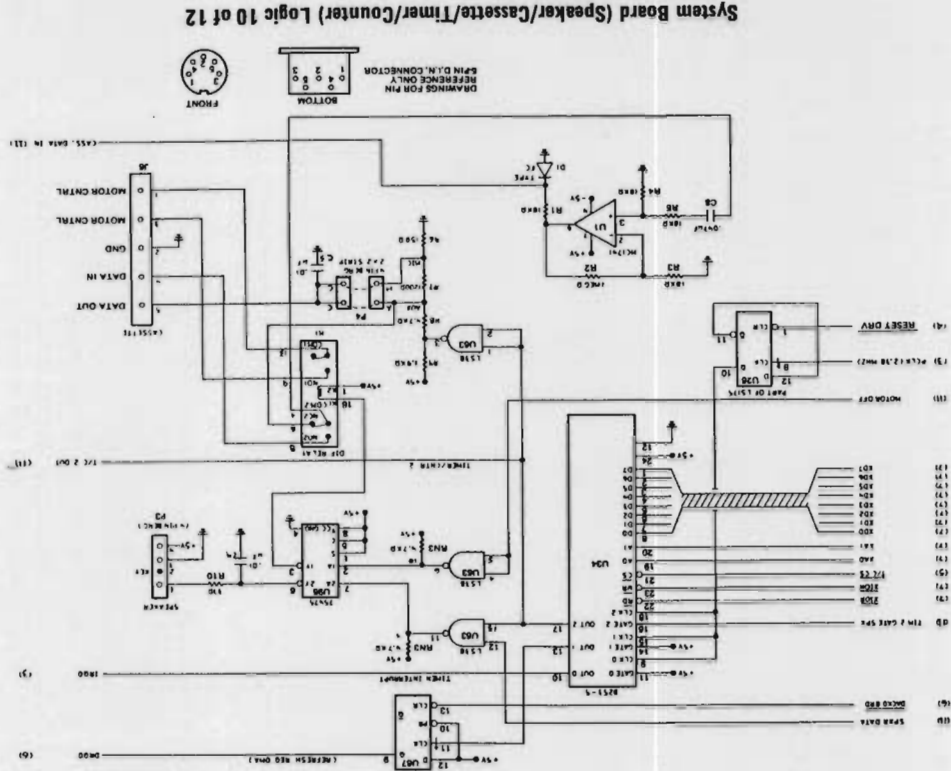
SYSTEM BOARD (DYNAMIC MEMORY)

System Board (ROS And Bus Driver) Logic 7 of 12



SYSTEM BOARD (ROS AND BUS DRIVER)

SYSTEM BOARD (SPEAKER/CASSETTE/TIMER/COUNTER)



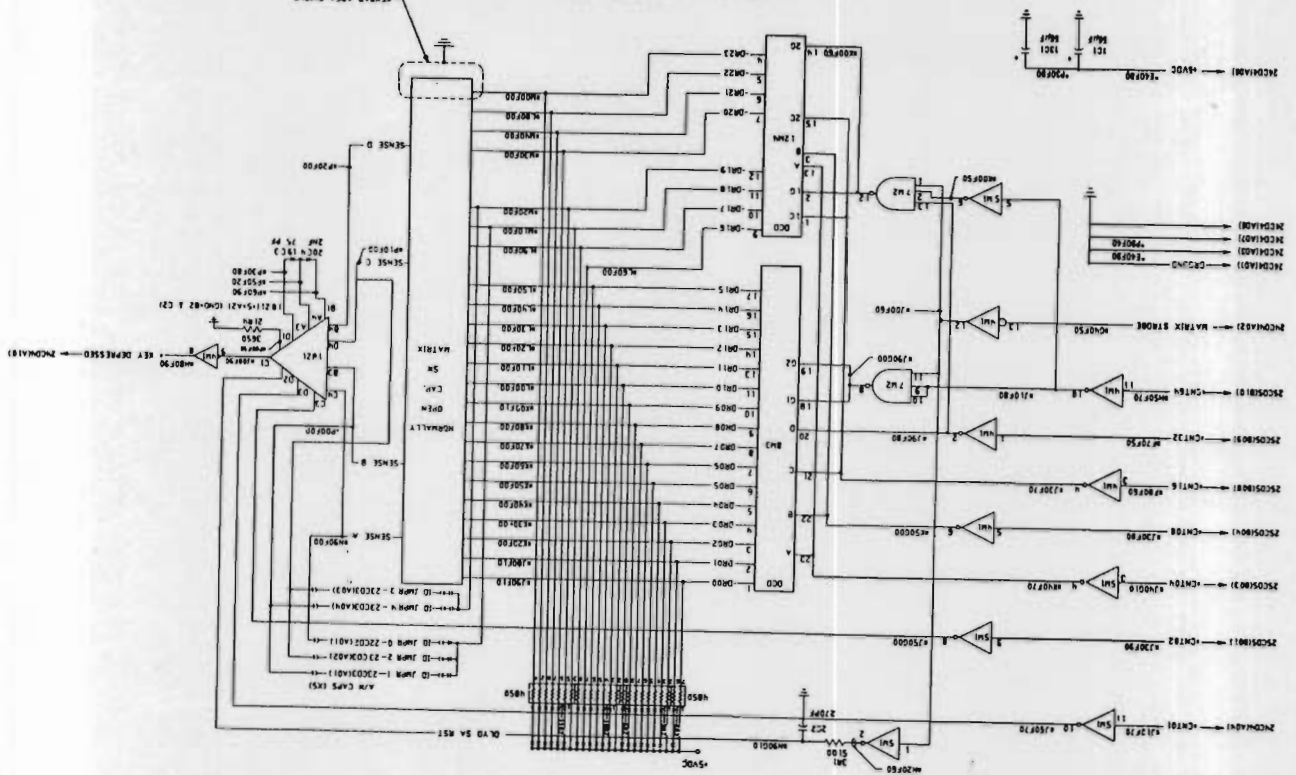
NOTE: ALL CAPS ARE B.277F TANTALUM ON THIS PAGE



D-10

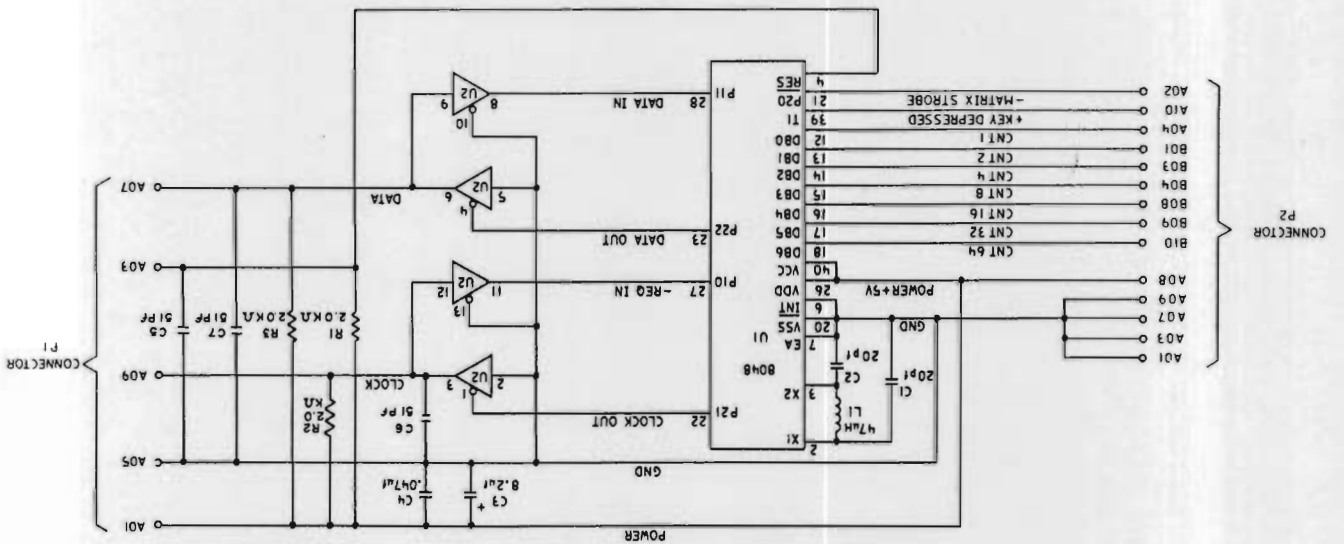


Keyboard Logic 2 of 2



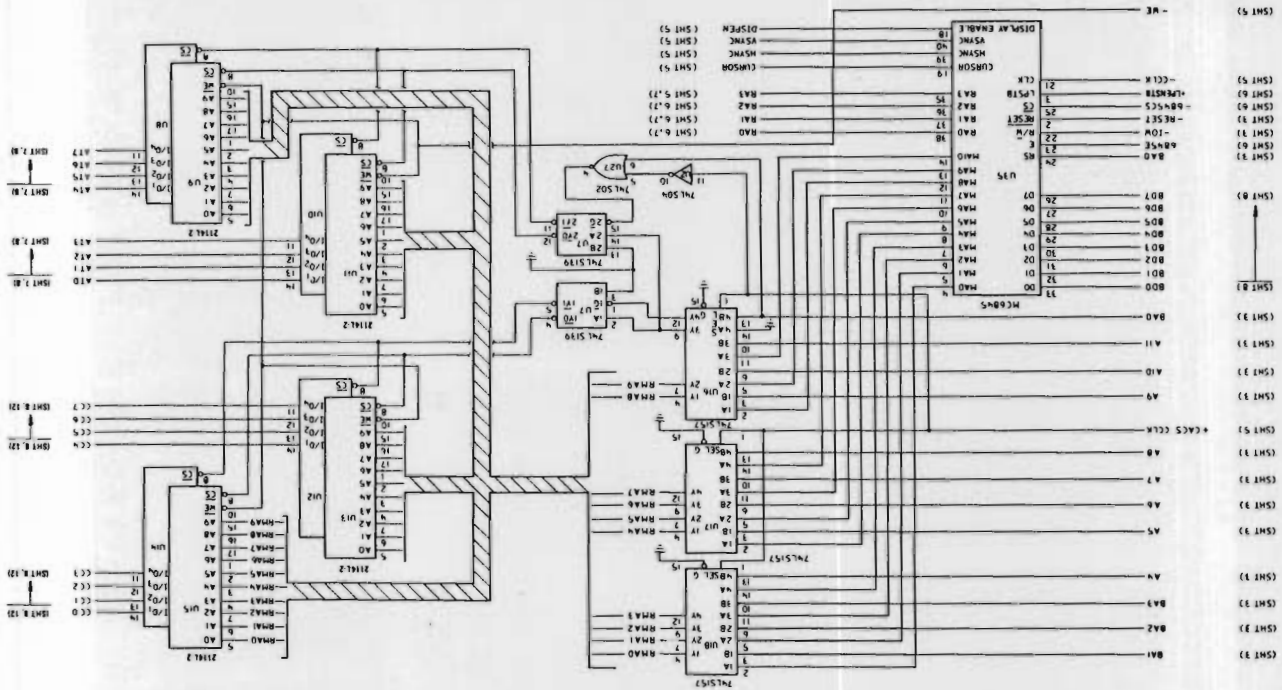
KEYBOARD

Keyboard Logic 1 of 2



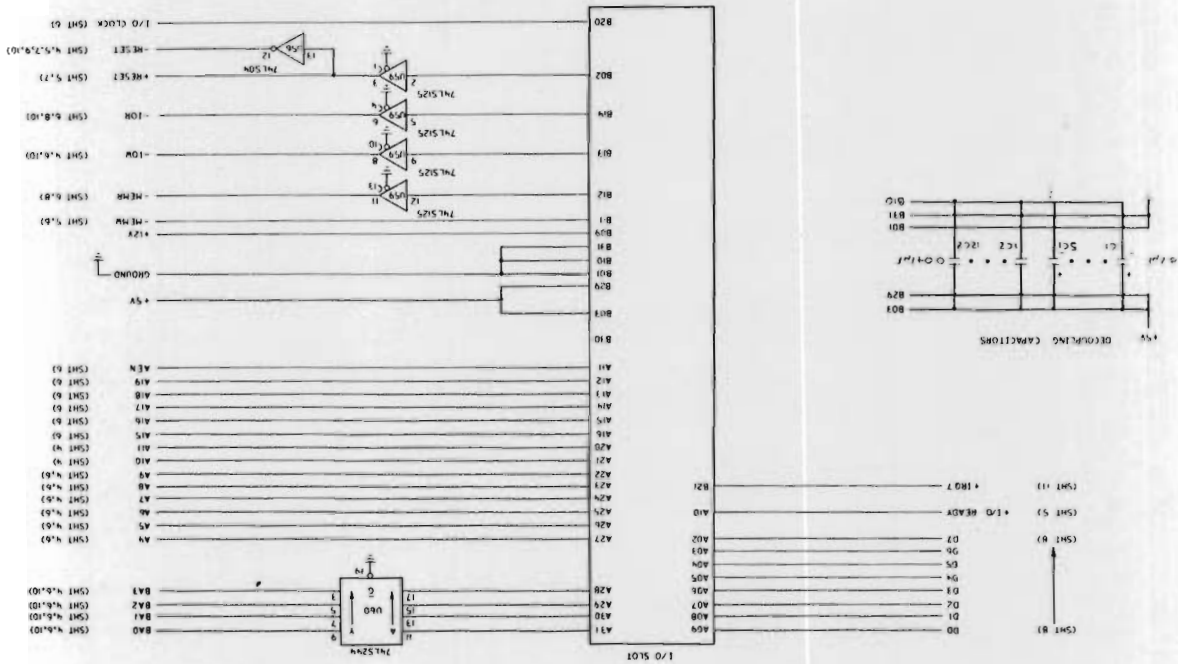
KEYBOARD

IBM MONOCHROME DISPLAY AND PARALLEL PRINTER ADAPTER



IBM Monochrome Display And Parallel Printer Adapter Logic 3 of 12

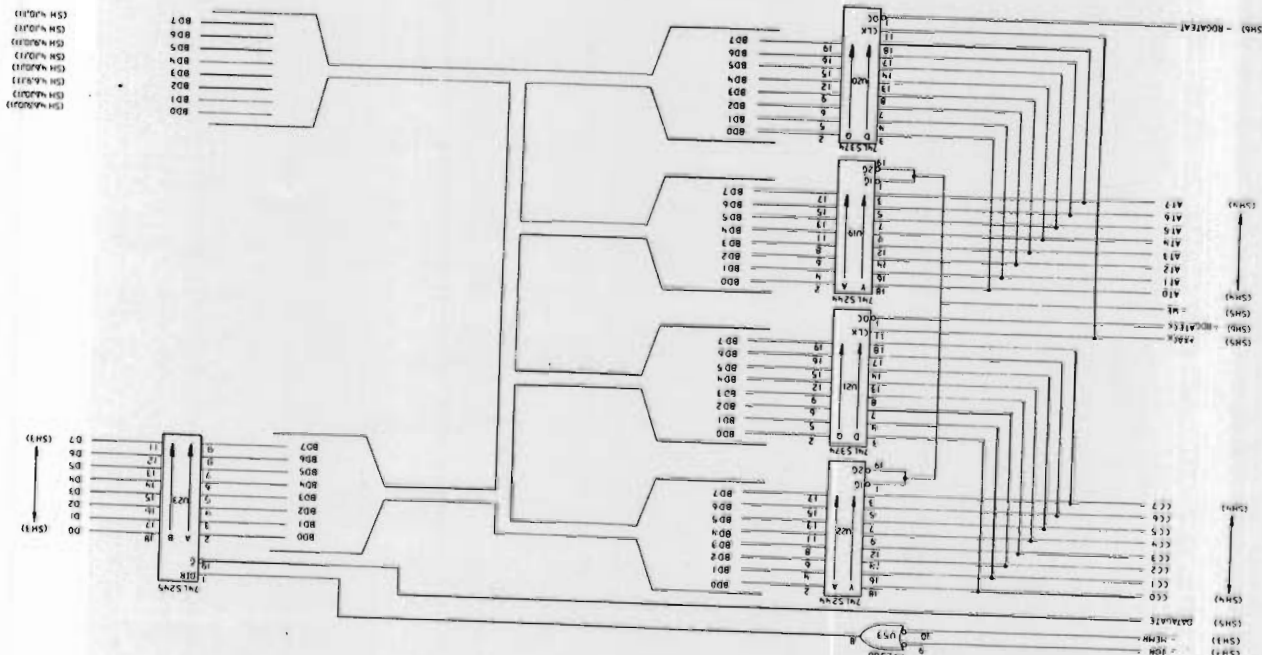
IBM MONOCHROME DISPLAY AND PARALLEL PRINTER ADAPTER



D-16

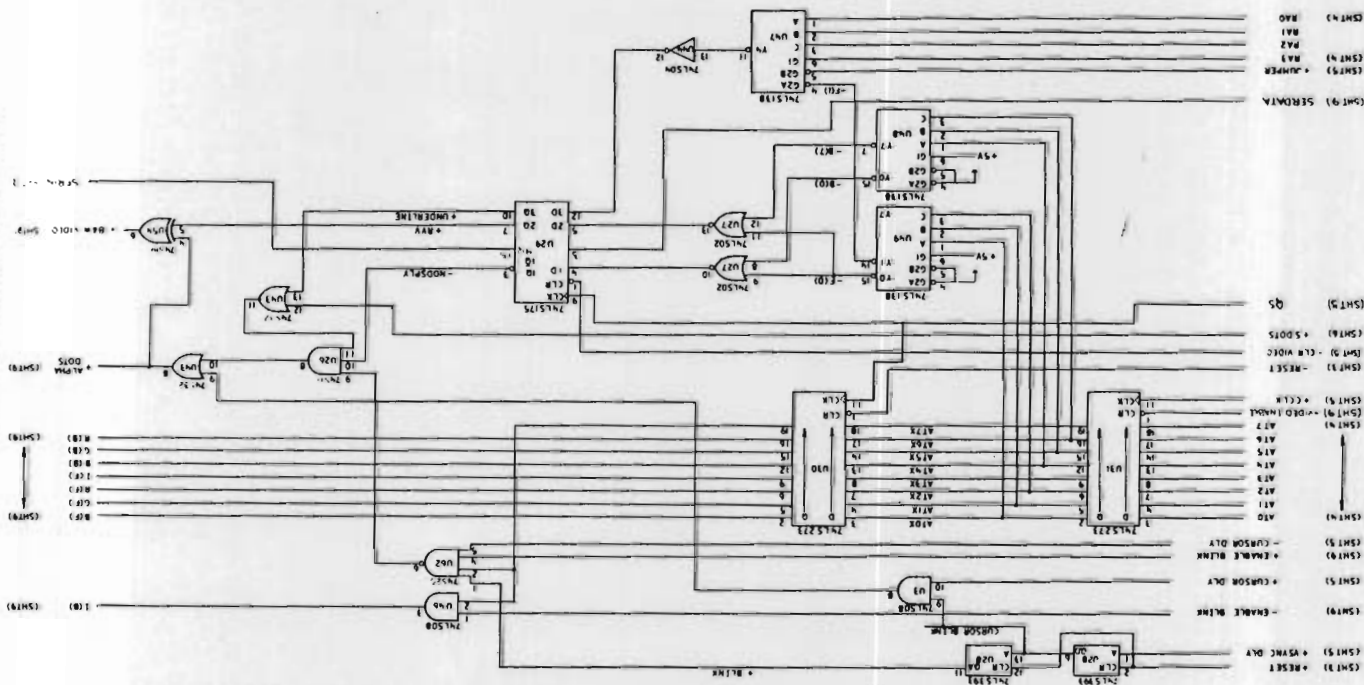
IBM MONOCHROME DISPLAY AND PARALLEL PRINTER ADAPTER

IBM MONOCHROME DISPLAY AND PARALLEL PRINTER ADAPTER

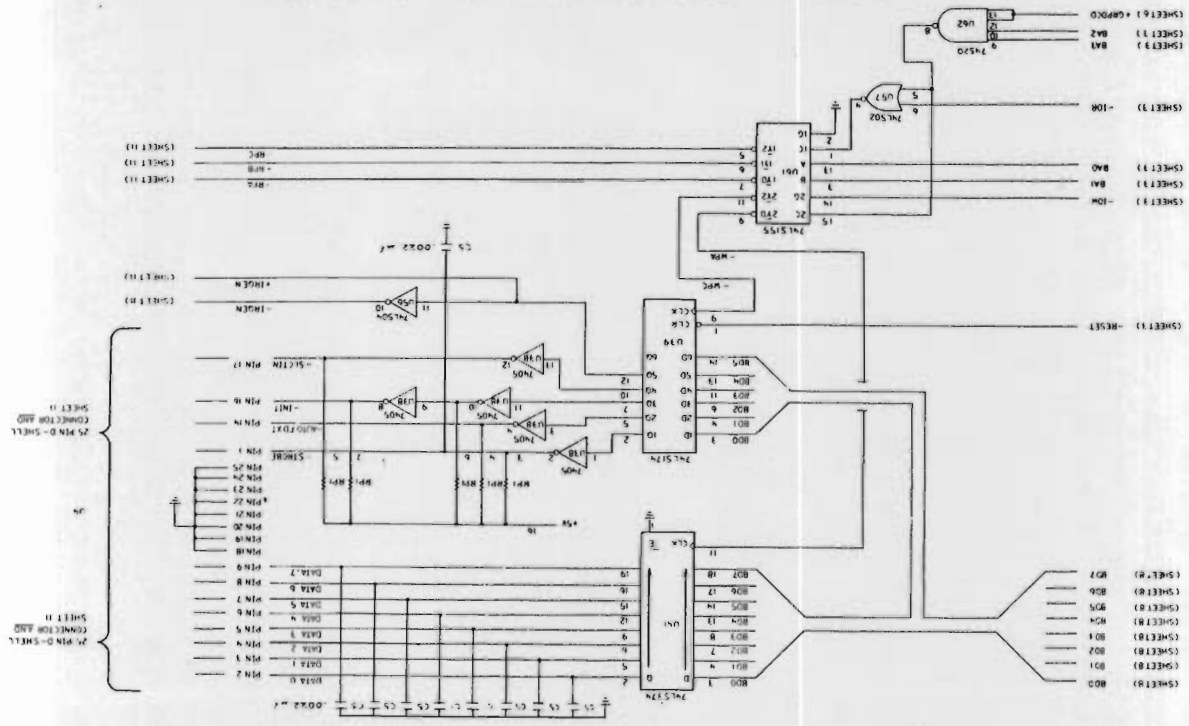


IBM Monochrome Display And Parallel Printer Adapter Logic 7 of 12

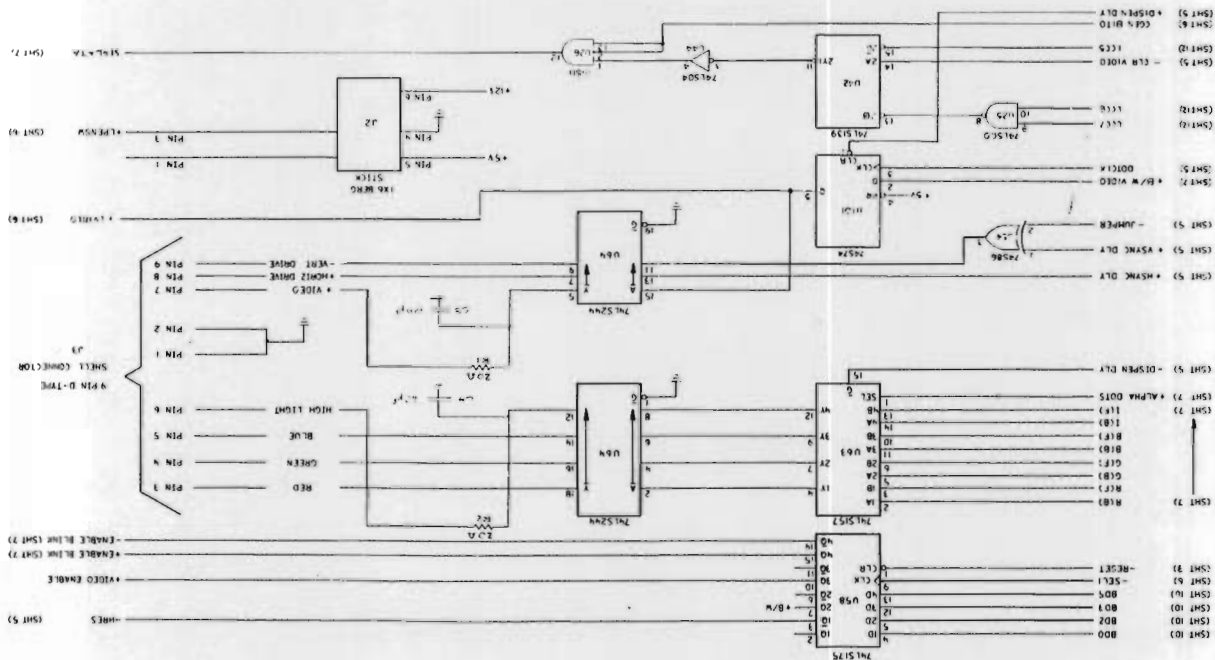
IBM MONOCHROME DISPLAY AND PARALLEL PRINTER ADAPTER



IBM Monochrome Display And Parallel Printer Adapter Logic 10 of 12



IBM Monochrome Display And Parallel Printer Adapter Logic 9 of 12

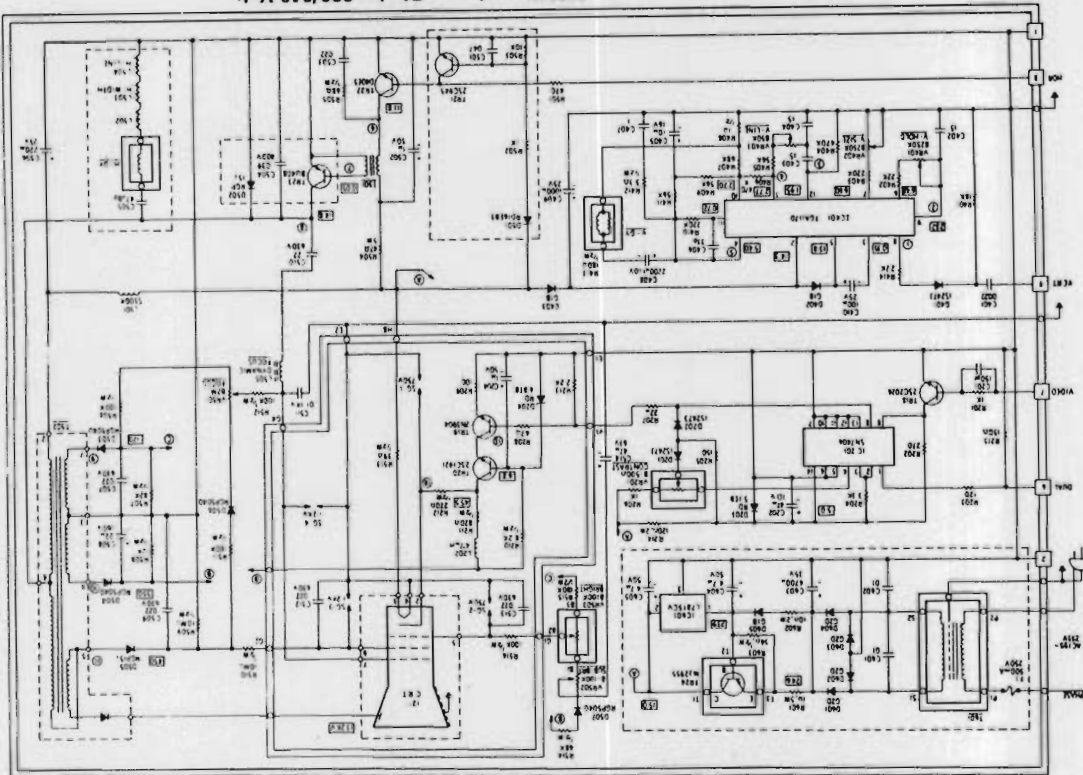


IBM Monochrome Display 220/240 Volt

NOTES

1. RESISTOR VALUES ARE IN OHMS UNLESS OTHERWISE SPECIFIED.
2. ALL CAPACITORS ARE POLYESTER UNLESS OTHERWISE SPECIFIED.
3. ALL CAPACITORS ARE POLYESTER UNLESS OTHERWISE SPECIFIED.
4. CAPACITORS ARE POLYESTER UNLESS OTHERWISE SPECIFIED.
5. CAPACITORS ARE POLYESTER UNLESS OTHERWISE SPECIFIED.
6. CAPACITORS ARE POLYESTER UNLESS OTHERWISE SPECIFIED.
7. CAPACITORS ARE POLYESTER UNLESS OTHERWISE SPECIFIED.
8. CAPACITORS ARE POLYESTER UNLESS OTHERWISE SPECIFIED.
9. CAPACITORS ARE POLYESTER UNLESS OTHERWISE SPECIFIED.
10. CAPACITORS ARE POLYESTER UNLESS OTHERWISE SPECIFIED.

WARNING: HAZARDOUS VOLTAGES UP TO 450 VOLTS EXIST ON THE PRINTED CIRCUIT BOARDS.



IBM Monochrome Display

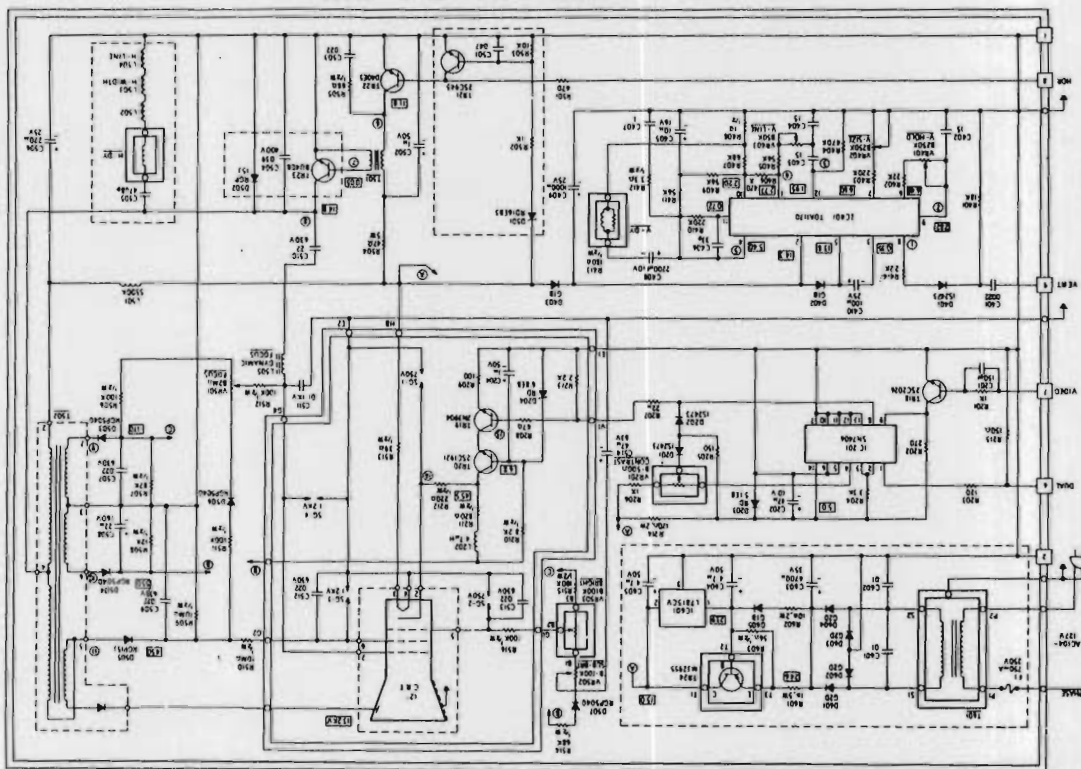
DANGER
HAZARDOUS VOLTAGES UP
TO 450 VOLTS EXIST ON THE
PRINTED CIRCUIT BOARDS

IBM Monochrome Display 120 Volt

NOTES

1. RESISTOR VALUES ARE IN OHMS UNLESS OTHERWISE SPECIFIED.
2. ALL CAPACITORS ARE POLYESTER UNLESS OTHERWISE SPECIFIED.
3. ALL CAPACITORS ARE POLYESTER UNLESS OTHERWISE SPECIFIED.
4. CAPACITORS ARE POLYESTER UNLESS OTHERWISE SPECIFIED.
5. CAPACITORS ARE POLYESTER UNLESS OTHERWISE SPECIFIED.
6. CAPACITORS ARE POLYESTER UNLESS OTHERWISE SPECIFIED.
7. CAPACITORS ARE POLYESTER UNLESS OTHERWISE SPECIFIED.
8. CAPACITORS ARE POLYESTER UNLESS OTHERWISE SPECIFIED.
9. CAPACITORS ARE POLYESTER UNLESS OTHERWISE SPECIFIED.
10. CAPACITORS ARE POLYESTER UNLESS OTHERWISE SPECIFIED.

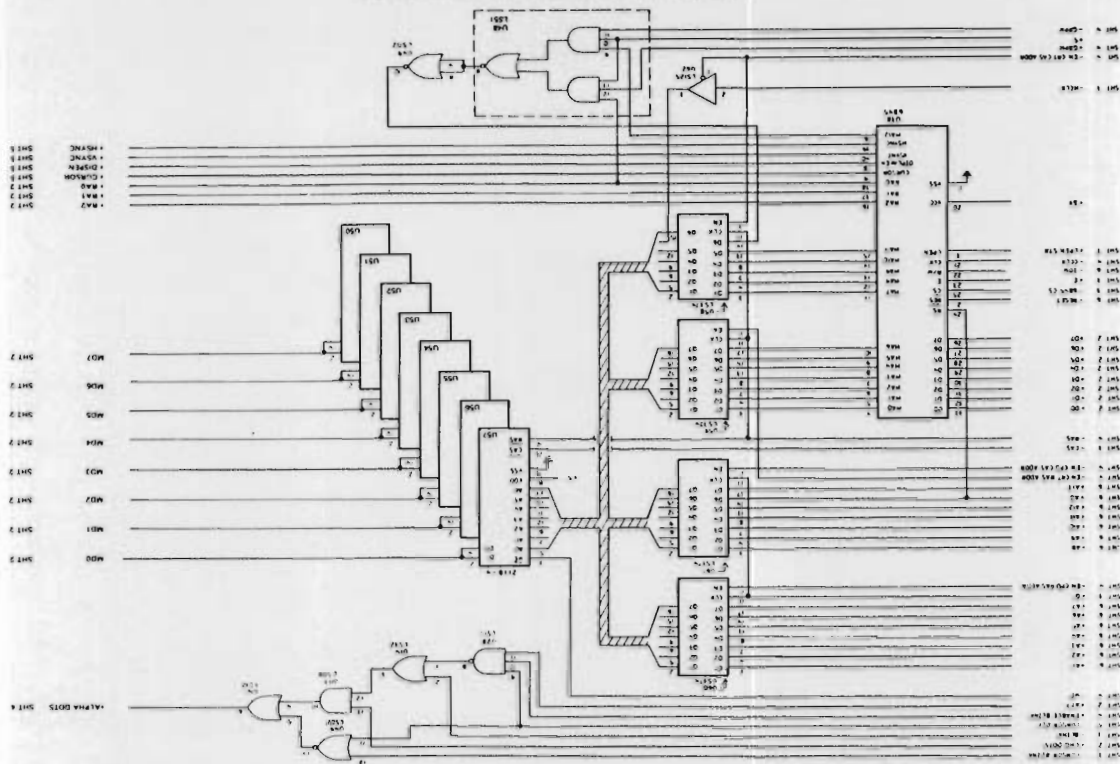
WARNING: HAZARDOUS VOLTAGES UP TO 450 VOLTS EXIST ON THE PRINTED CIRCUIT BOARDS.



IBM MONOCHROME DISPLAY

DANGER
HAZARDOUS VOLTAGES UP
TO 450 VOLTS EXIST ON THE
PRINTED CIRCUIT BOARDS

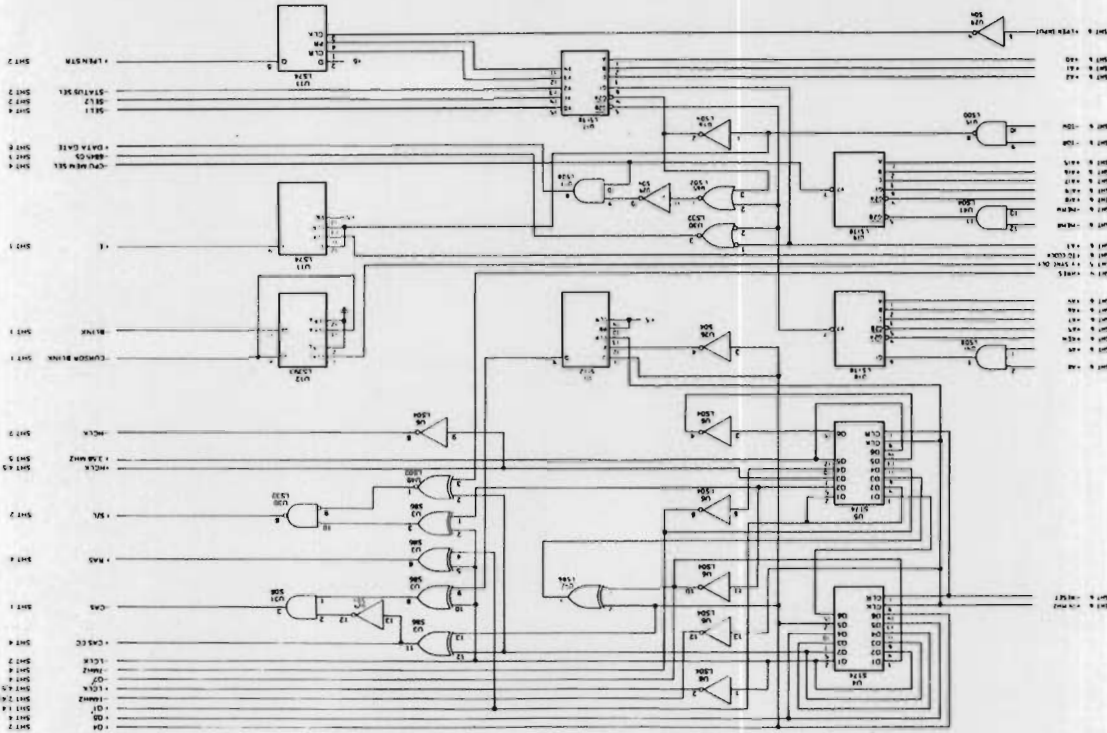
COLOR/GRAPHICS MONITOR ADAPTER



Color/Graphics Monitor Adapter Logic 1 of 6

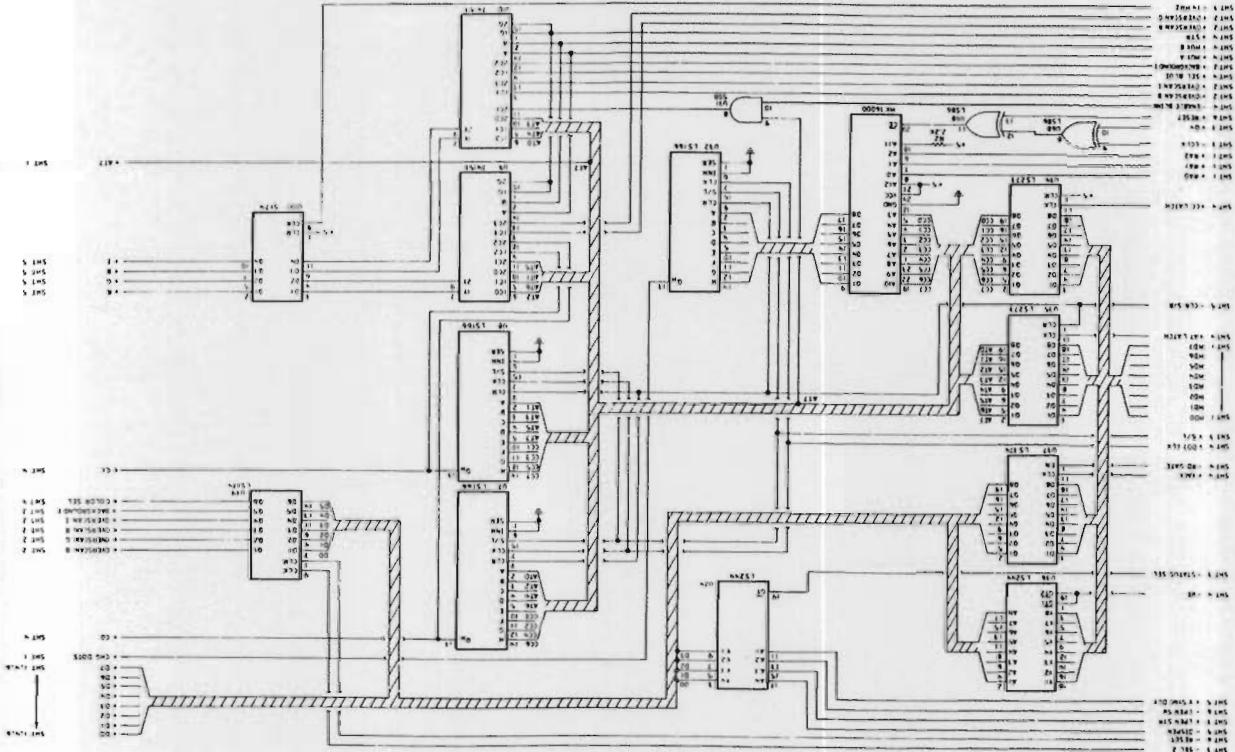
Color/Graphics Monitor Adapter Logic 3 of 6

COLOR/GRAPHICS MONITOR ADAPTER



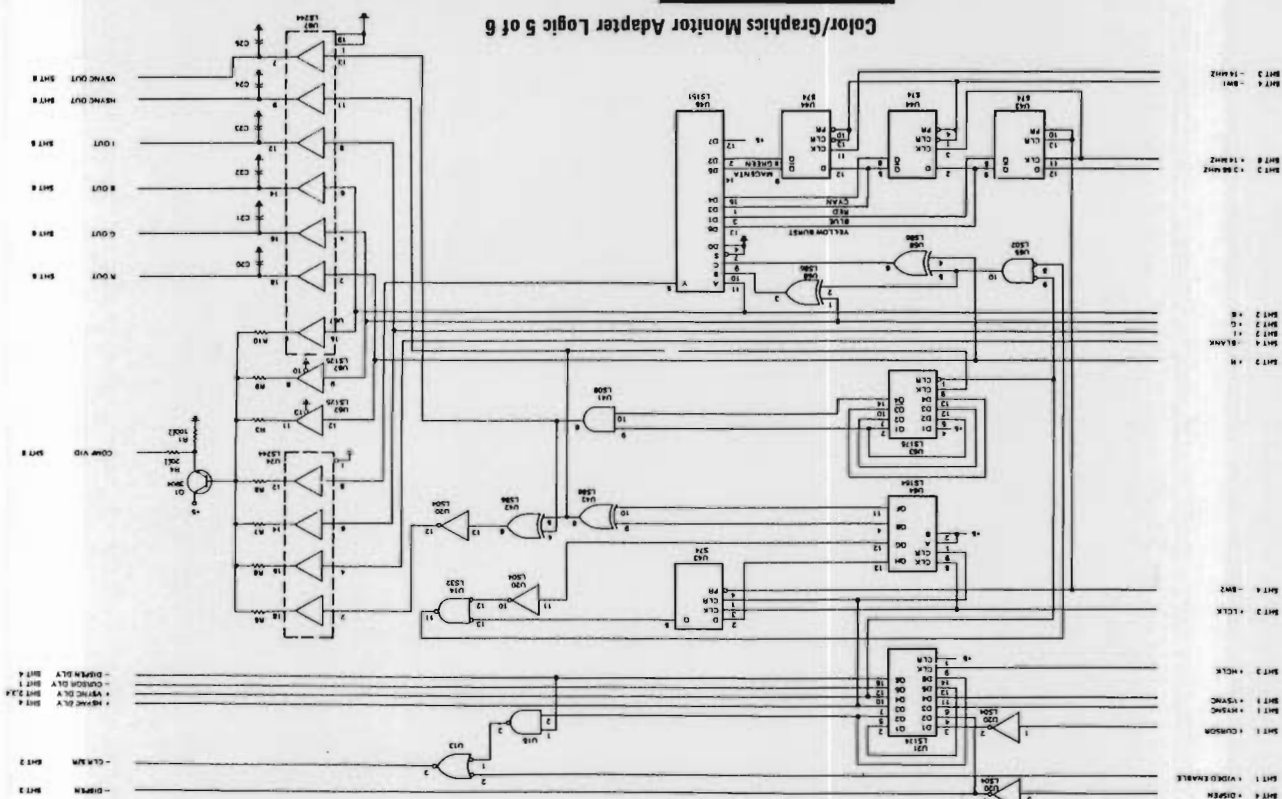
Color/Graphics Monitor Adapter Logic 2 of 6

COLOR/GRAPHICS MONITOR ADAPTER



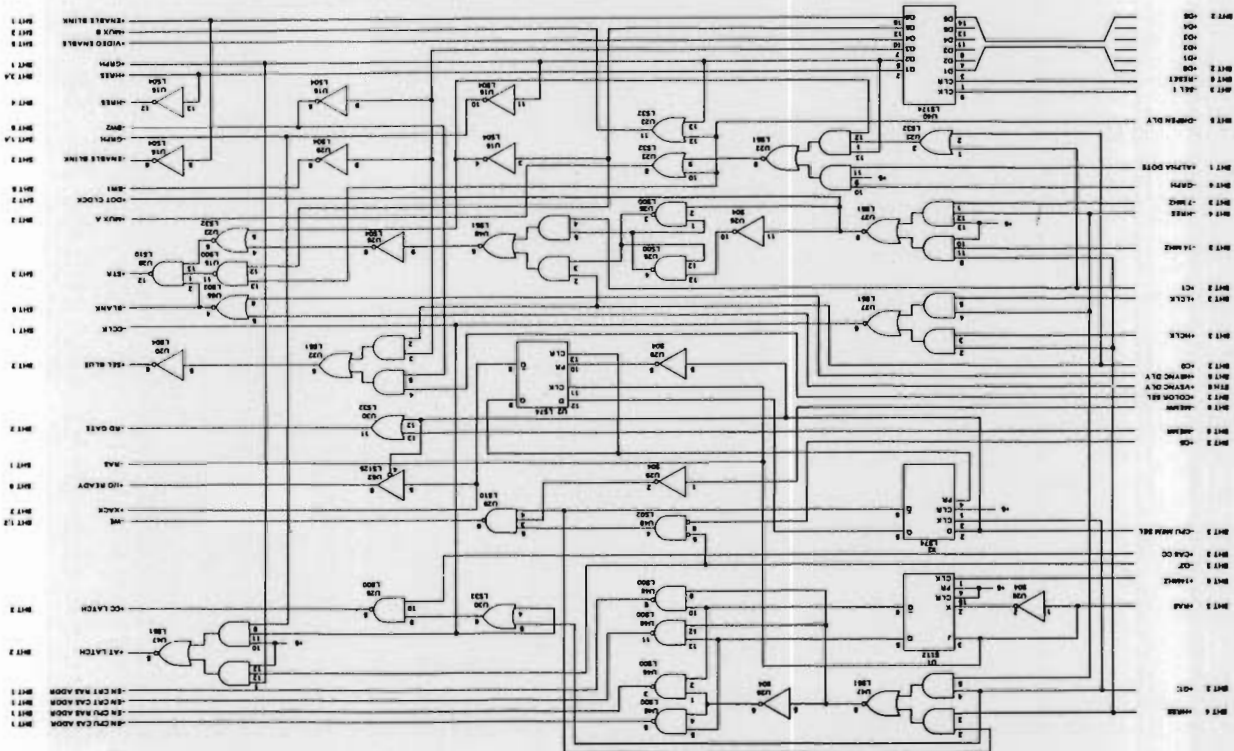
Color/Graphics Monitor Adapter Logic 5 of 6

COLOR/GRAPHICS MONITOR ADAPTER



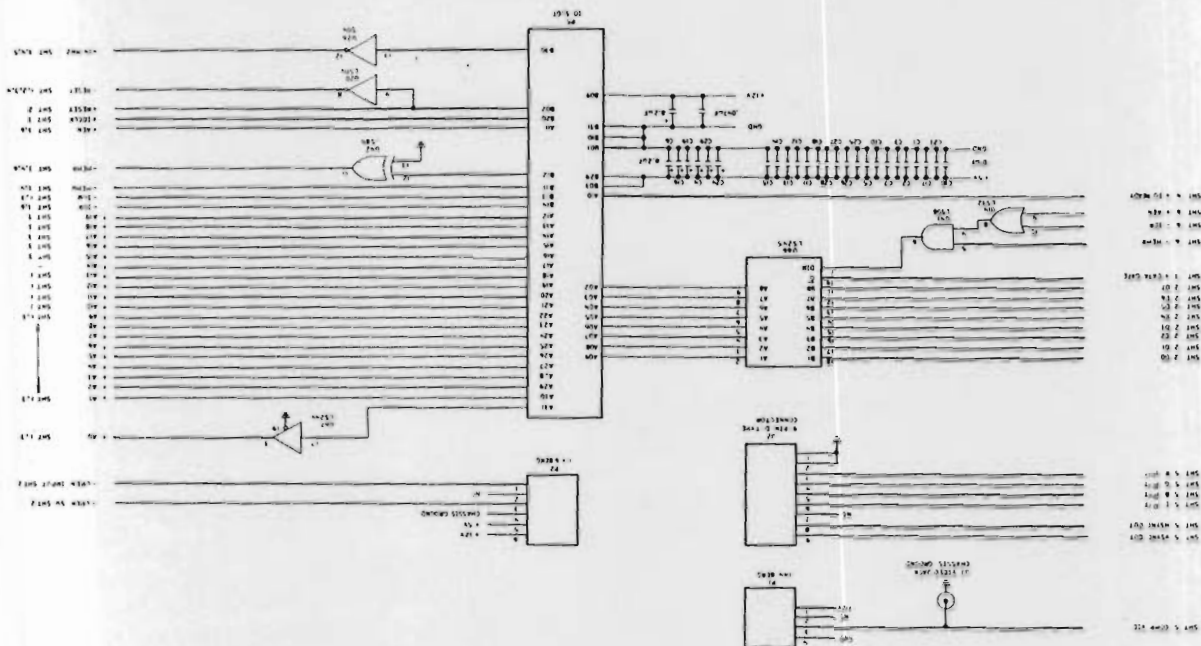
Color/Graphics Monitor Adapter Logic 4 of 6

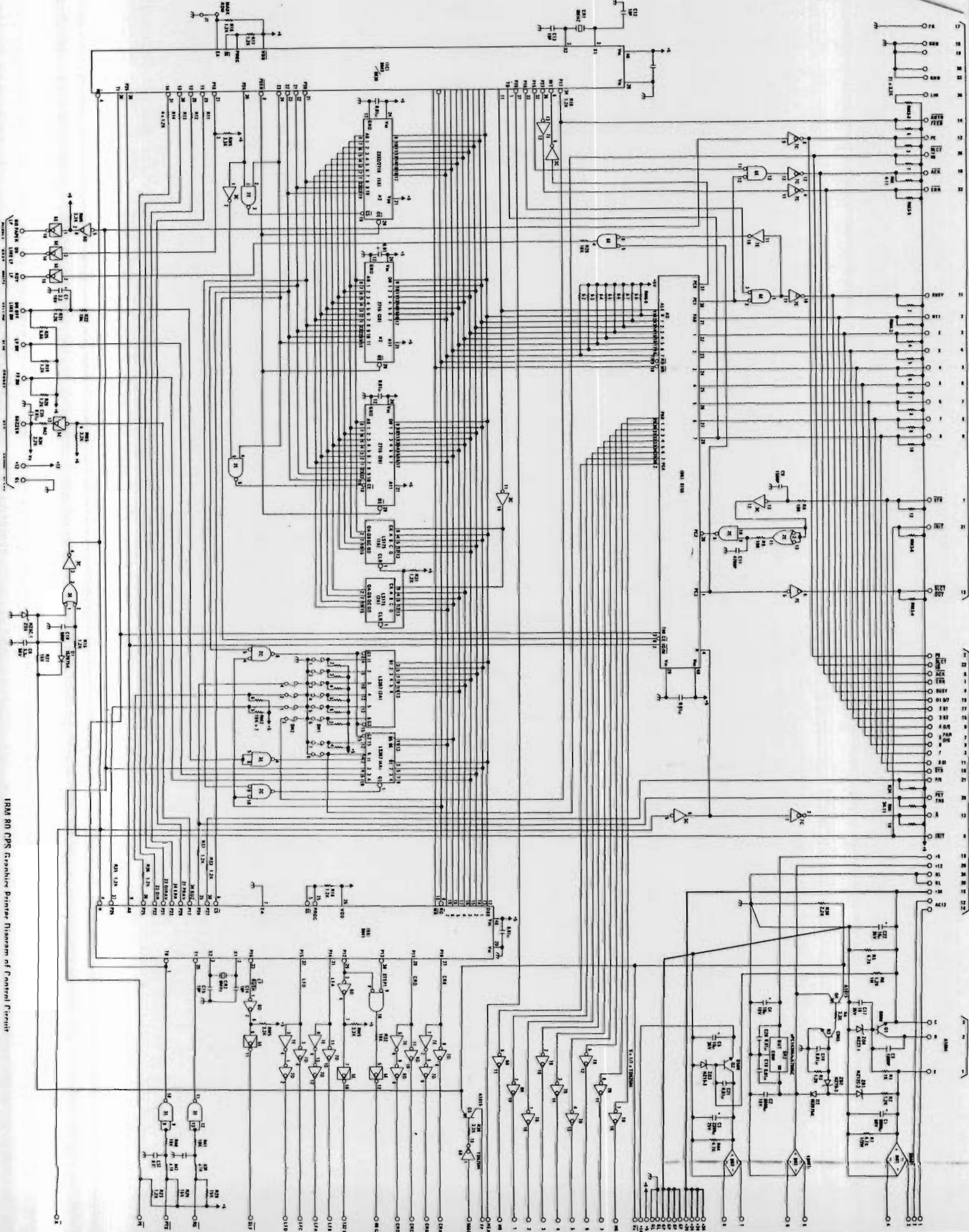
COLOR/GRAPHICS MONITOR ADAPTER



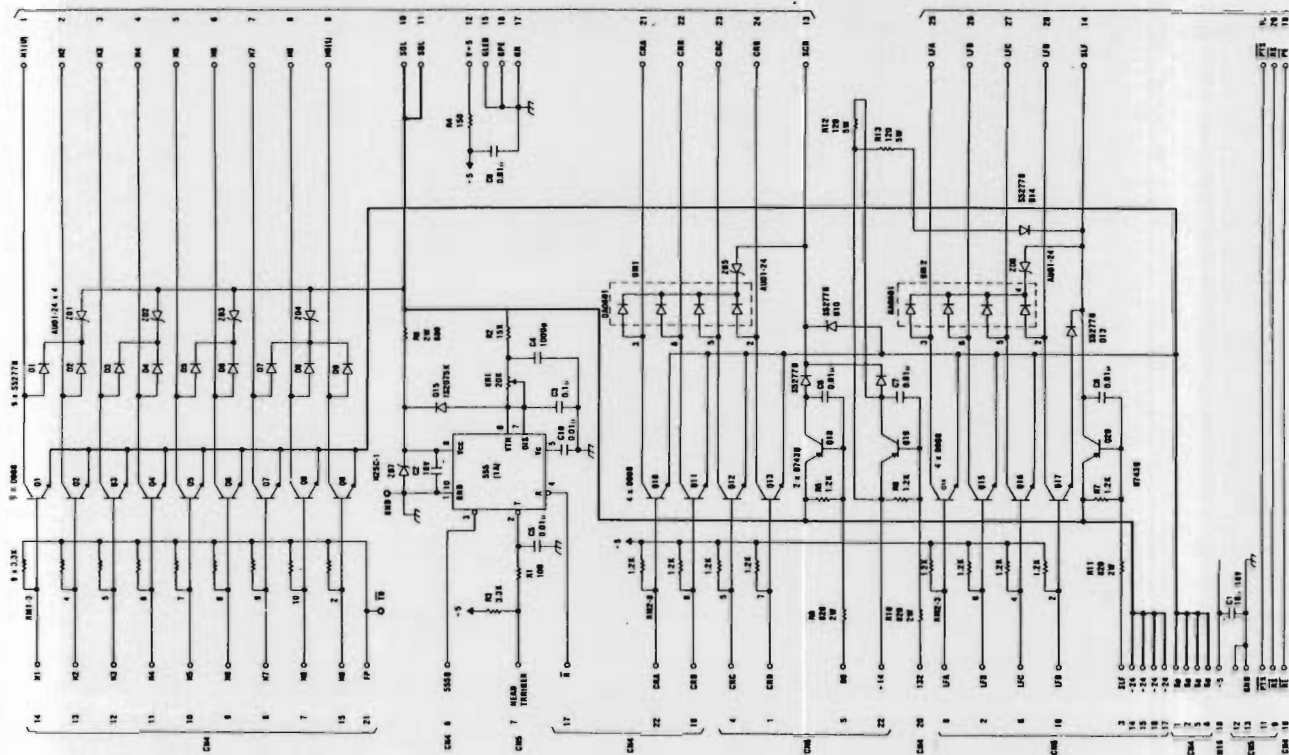
COLOR/GRAPHICS MONITOR ADAPTER

Color/Graphics Monitor Adapter Logic 6 of 6

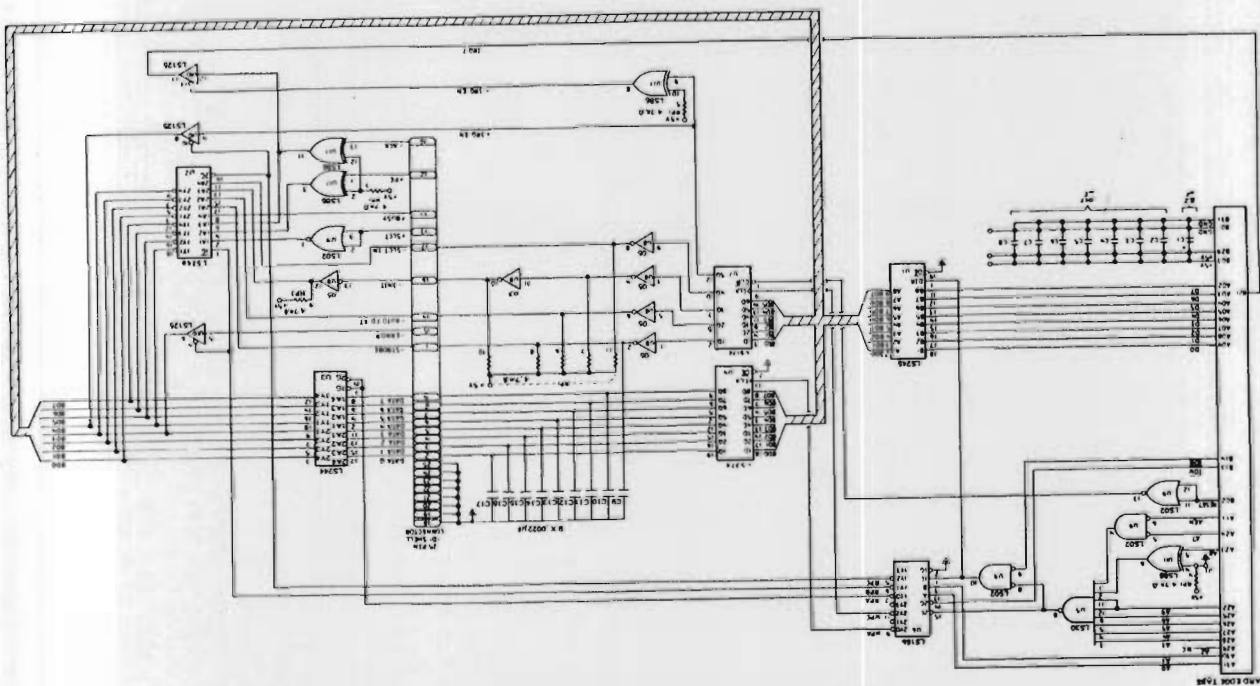




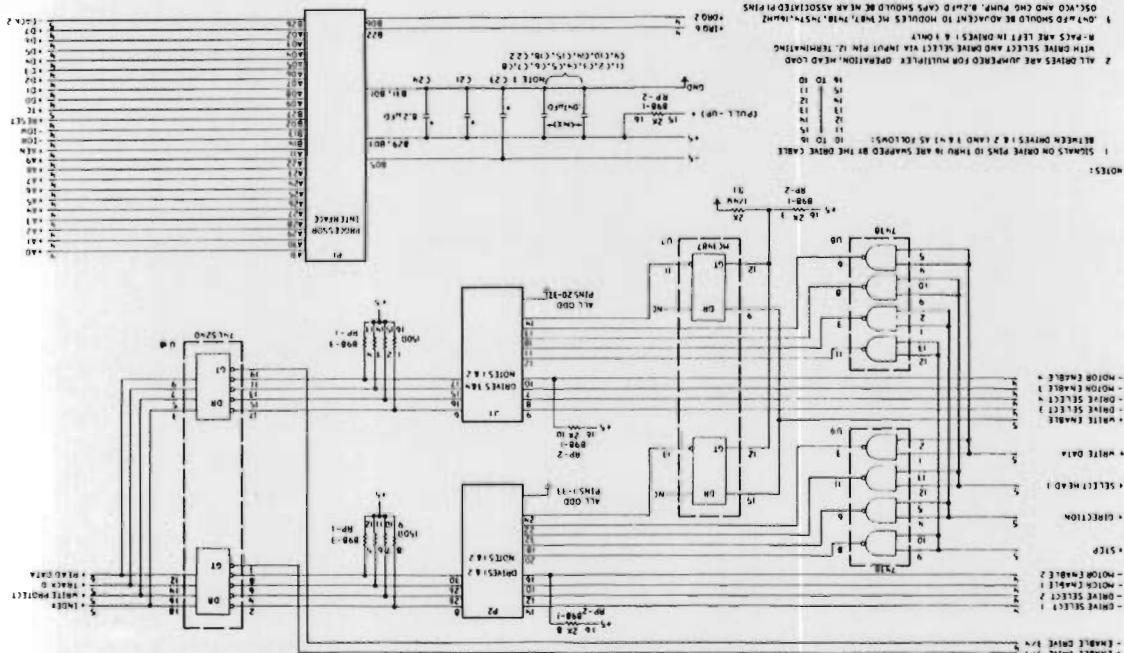
IBM 80 CPS DIAGRAM OF DRIVER CIRCUIT GRAPHICS PRINTER



PARALLEL PRINTER ADAPTER



5 1/4" DISKETTE DRIVE ADAPTER



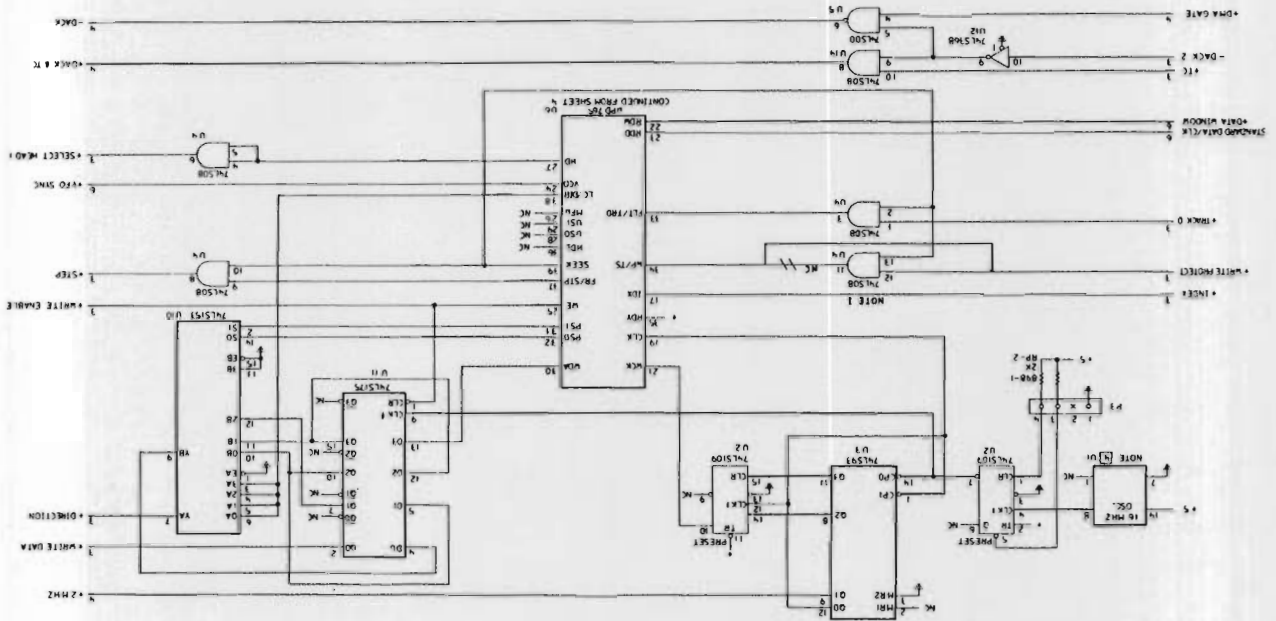
Note: Logics one and two of six are not applicable.

5 1/4" Diskette Drive Adapter Logic 3 of 6

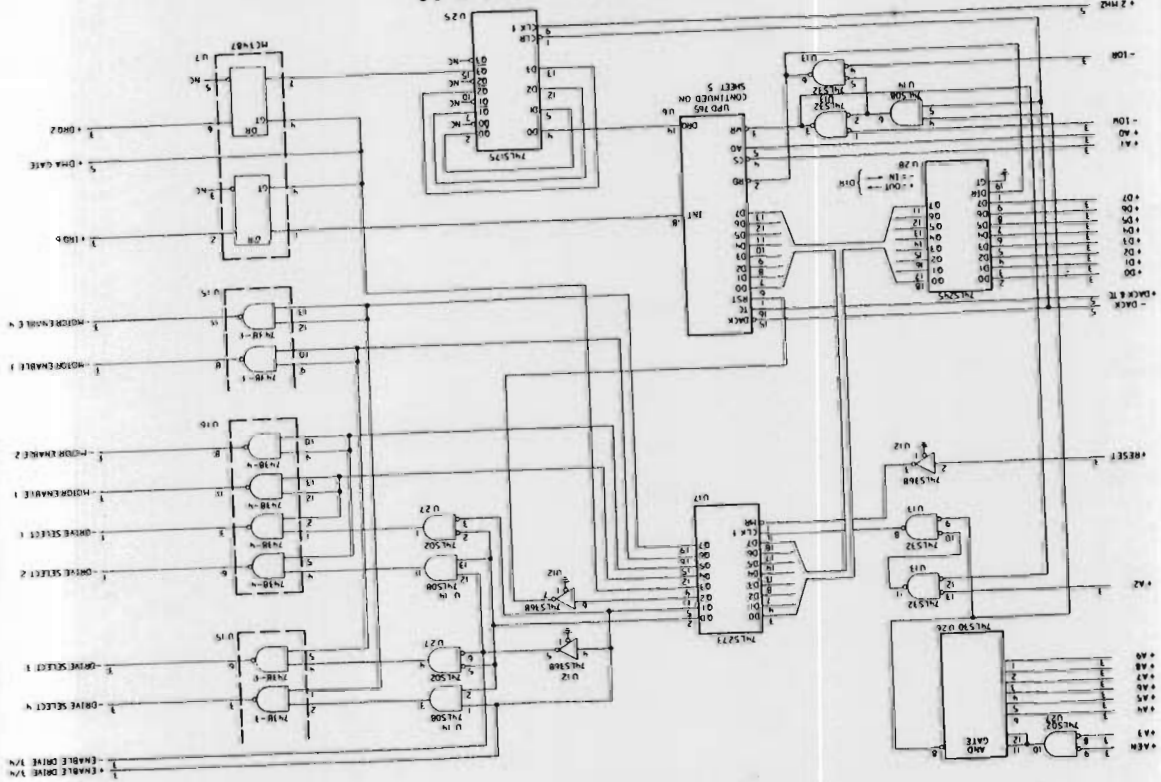
- [illegible]

5 1/4" Diskette Drive Adapter Logic 5 of 6

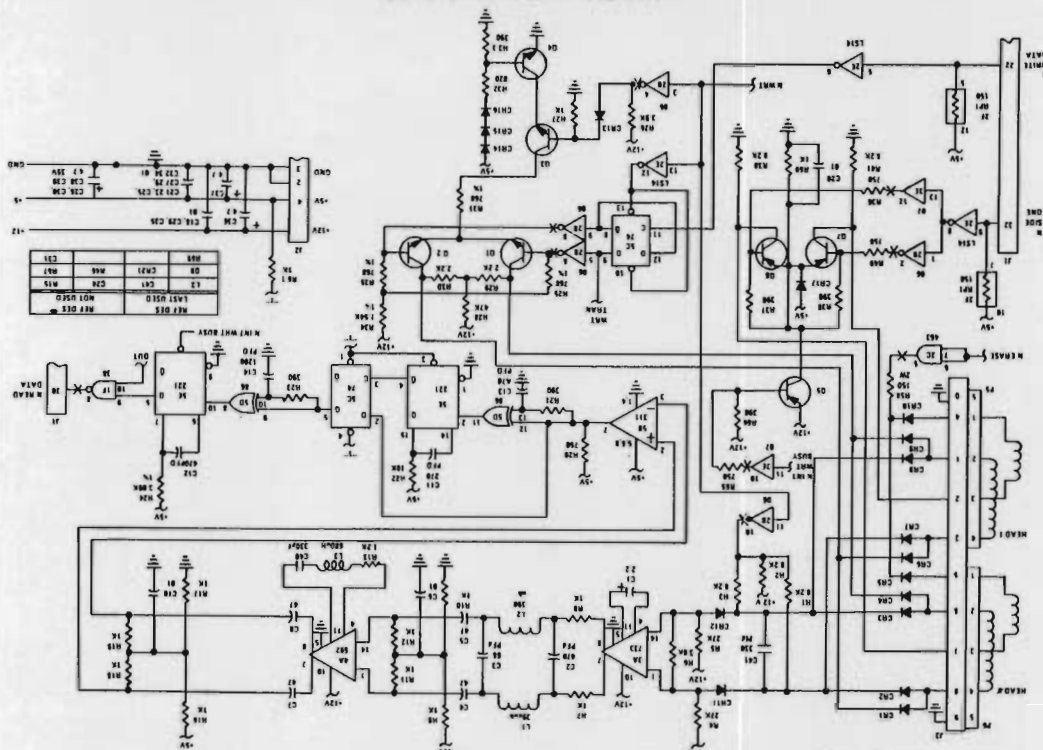
NOTE: U4 (74LS00) PINS 12 AND 13 ARE CONNECTED ONLY ON CARDS
BUILT LISTING MAY CARD IN 5001293



5% "Diskette Drive Adapter Logic 4 of 6

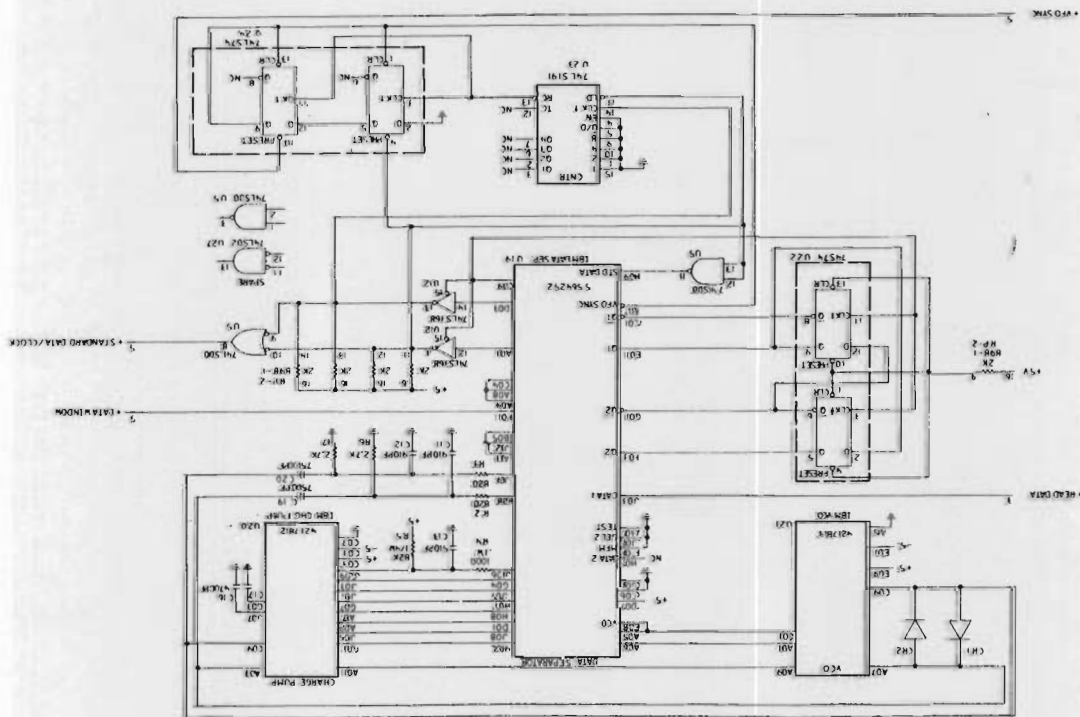


5 1/4" Diskette Drive Logic 1 of 3



5 1/4" DISKETTE DRIVE

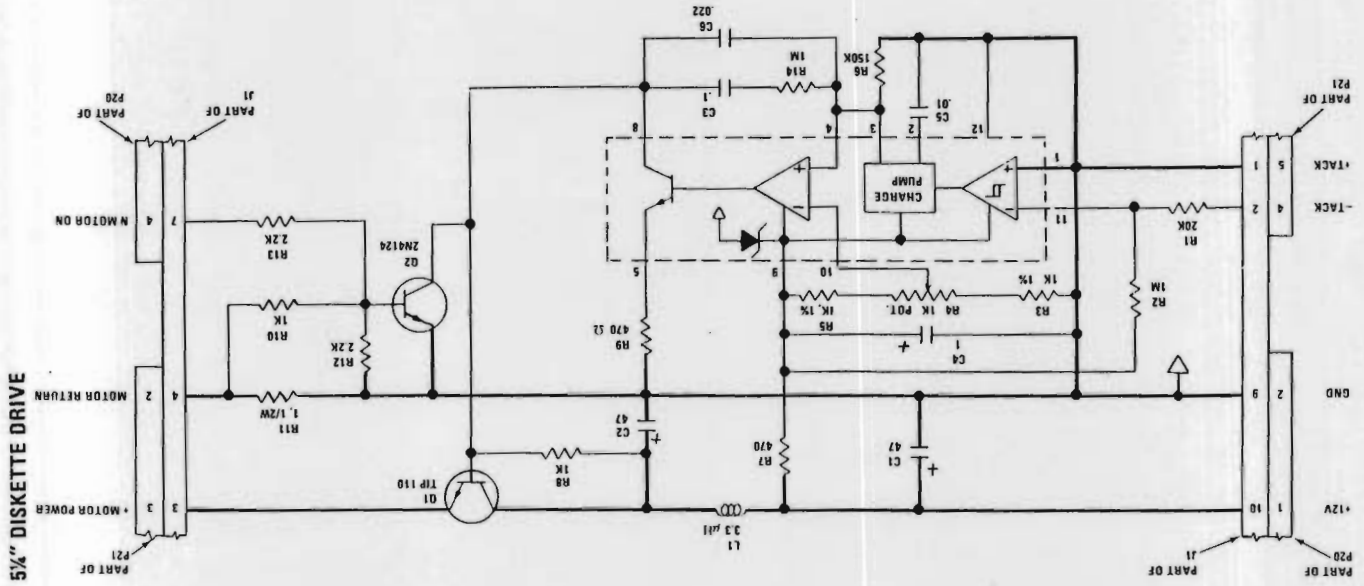
5 1/4" Diskette Drive Adapter Logic 6 of 6



5 1/4" DISKETTE DRIVE ADAPTER

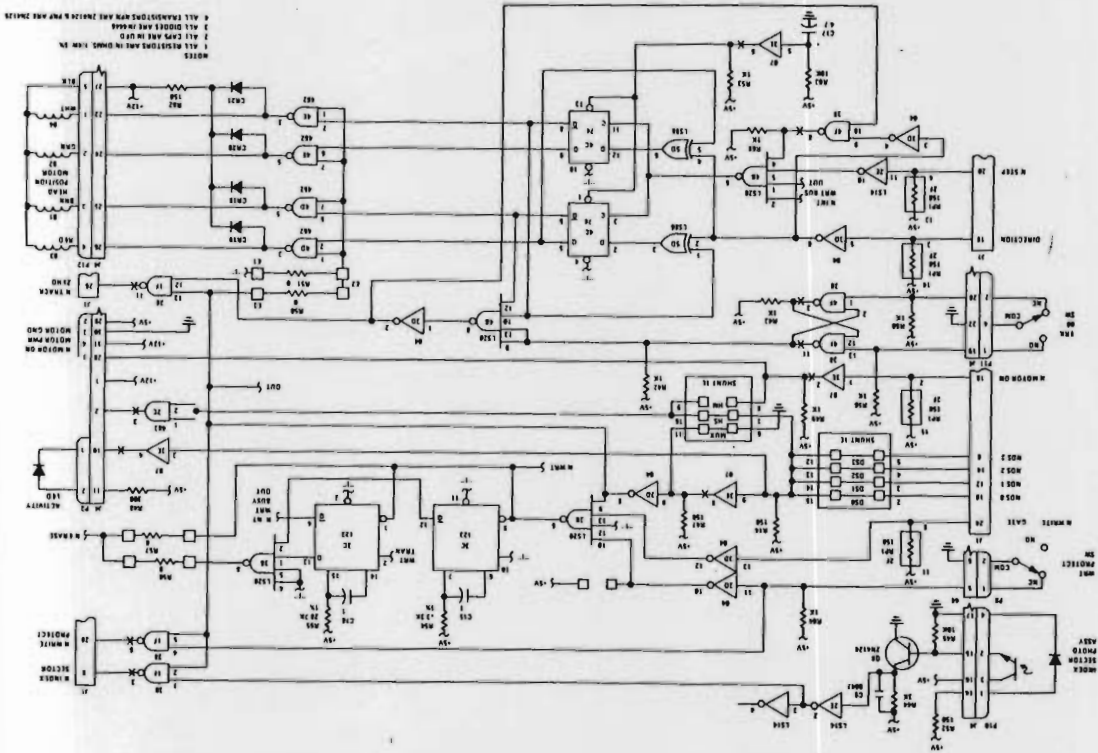
5 1/4" Diskette Drive Logic 3 of 3

NOTES:
1. RESISTORS ARE IN OHMS, ± 5%, 1/4 W.
2. 1% RESISTORS ARE 1/8 W.
3. CAPACITORS ARE IN μ F, ± 20%, 35 V.



5 1/4" DISKETTE DRIVE

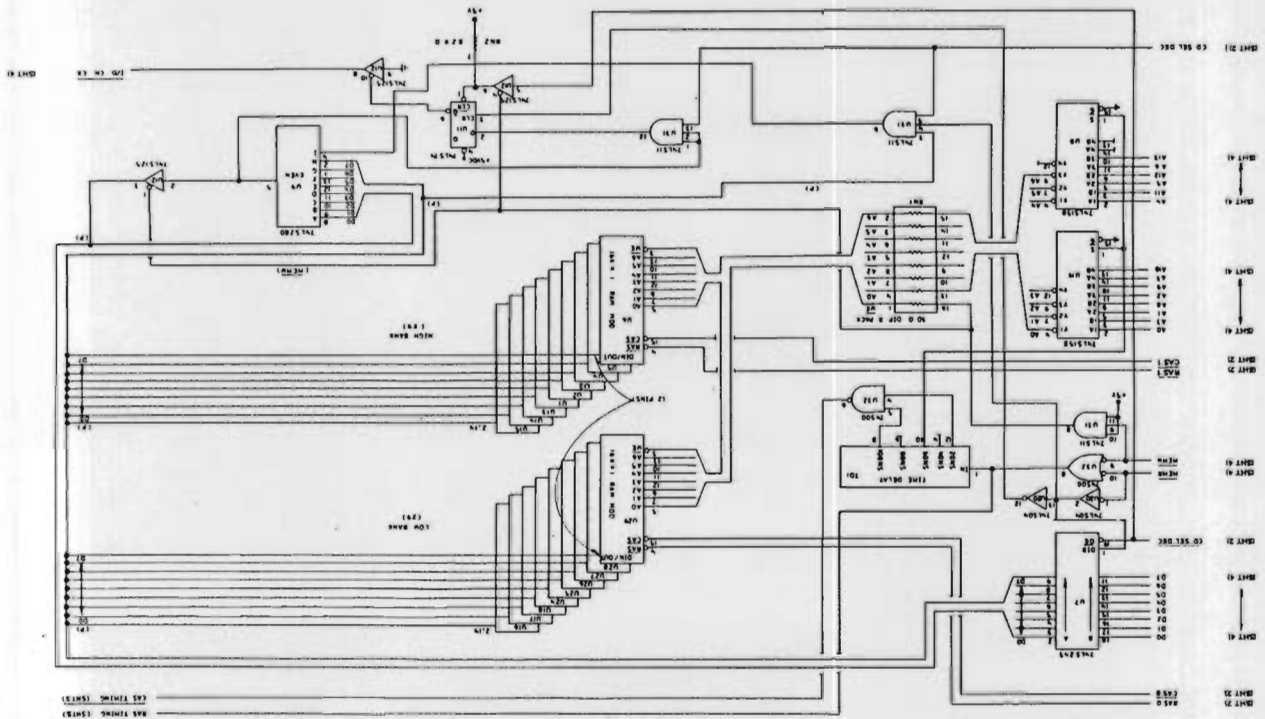
5 1/4" Diskette Drive Logic 2 of 3



5 1/4" DISKETTE DRIVE

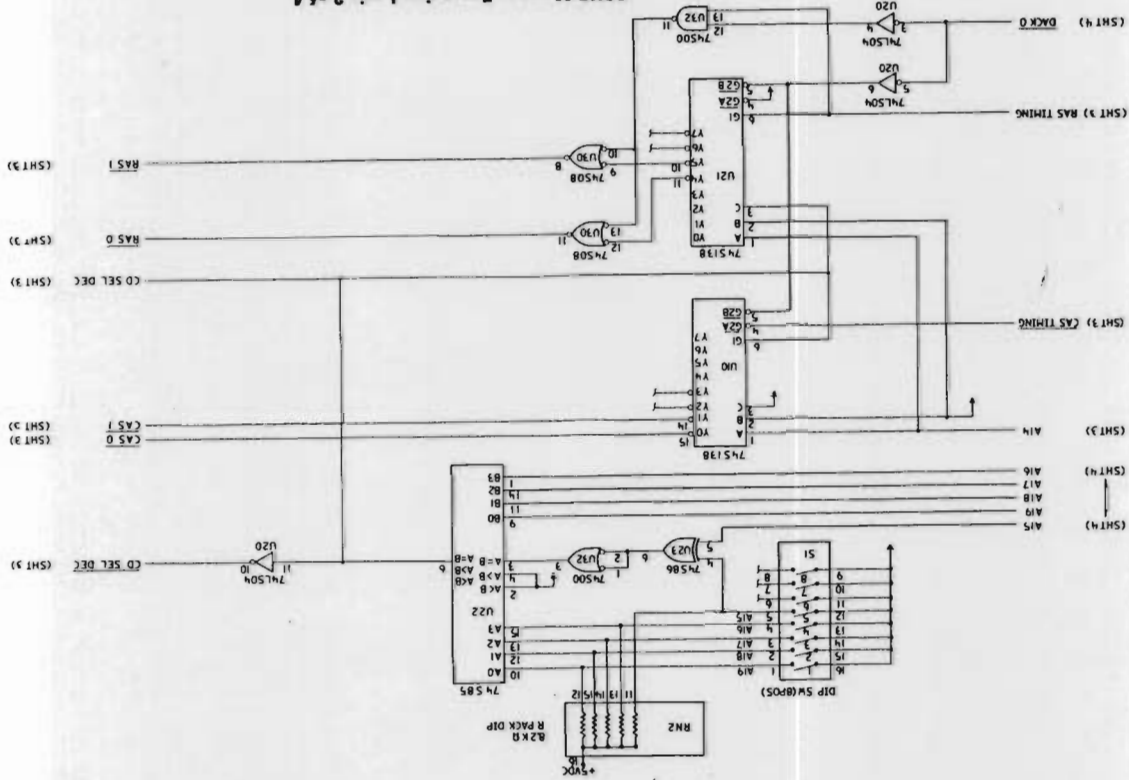
32KB Memory Expansion Logic 3 of 4

32 KB MEMORY EXPANSION

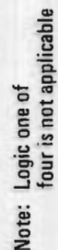


32KB Memory Expansion Logic 2 of 4

32 KB MEMORY EXPANSION



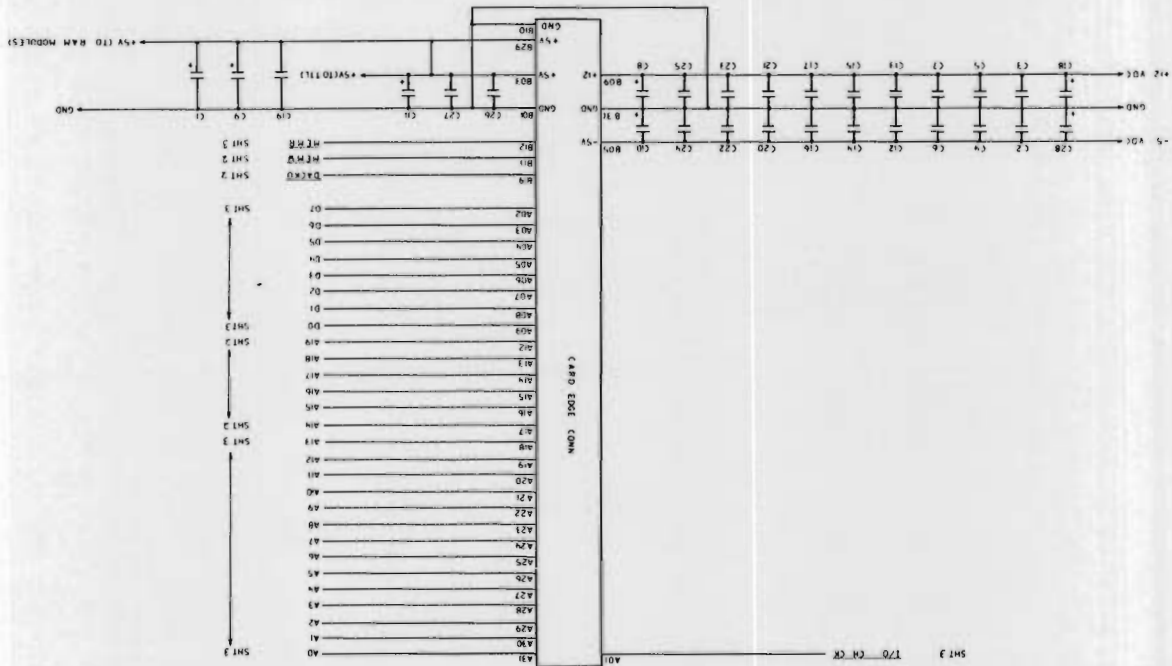
Note: Logic one of four is not applicable



32KB MEMORY EXPANSION

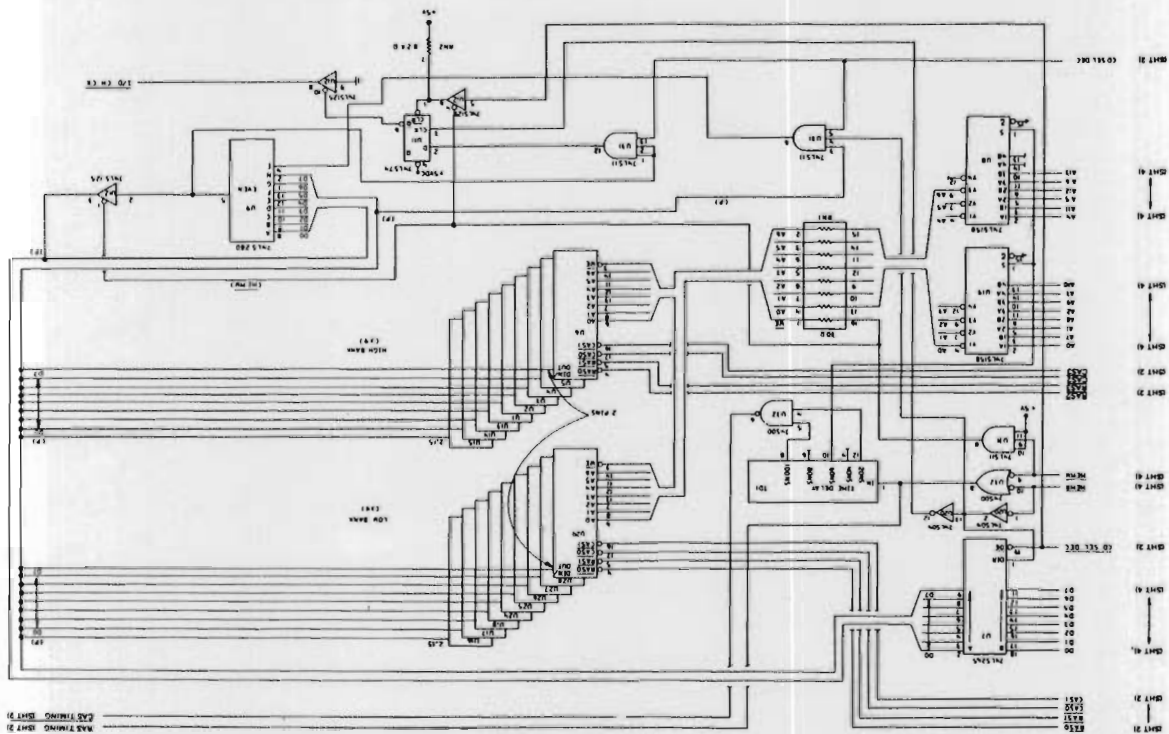


64 KB Memory Expansion Logic 4 of 4

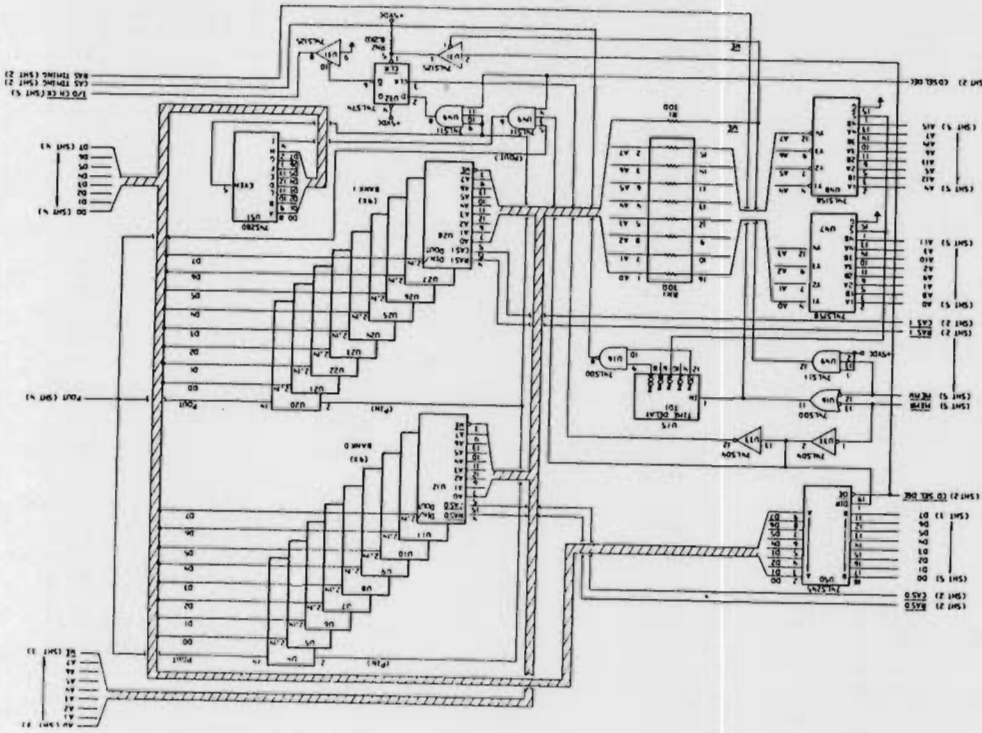


64 KB MEMORY EXPANSION

64KB Memory Expansion Logic 3 of 4

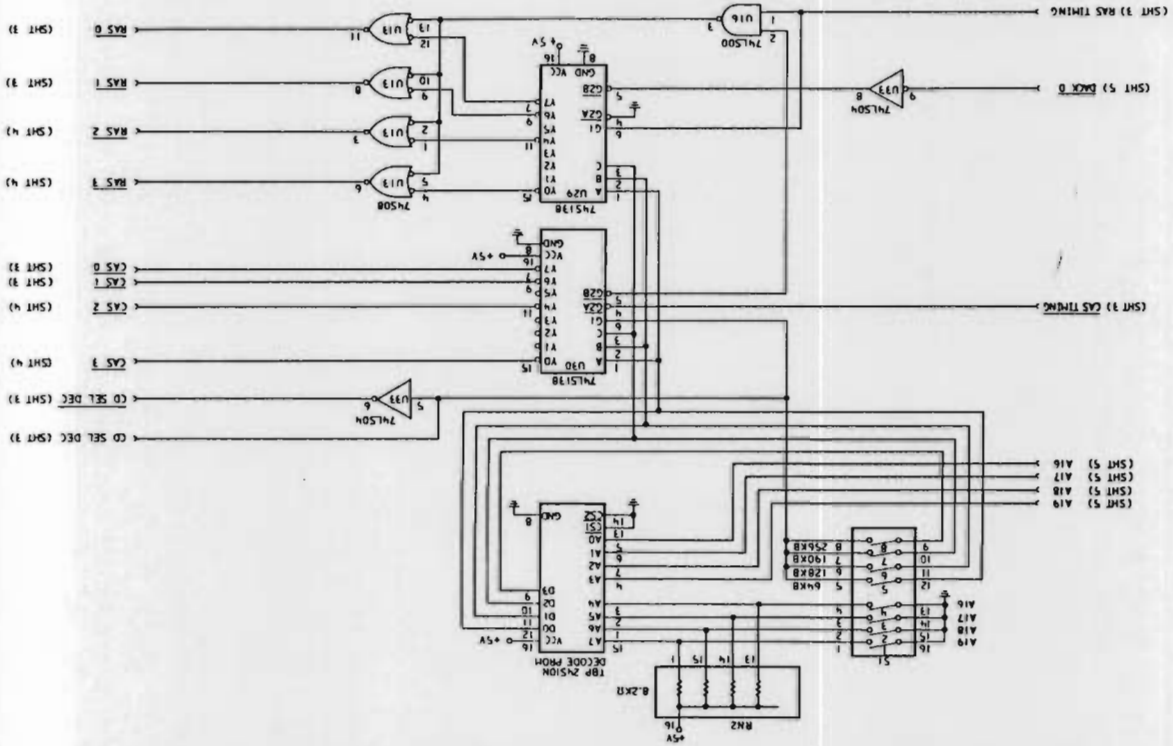


64 KB MEMORY EXPANSION



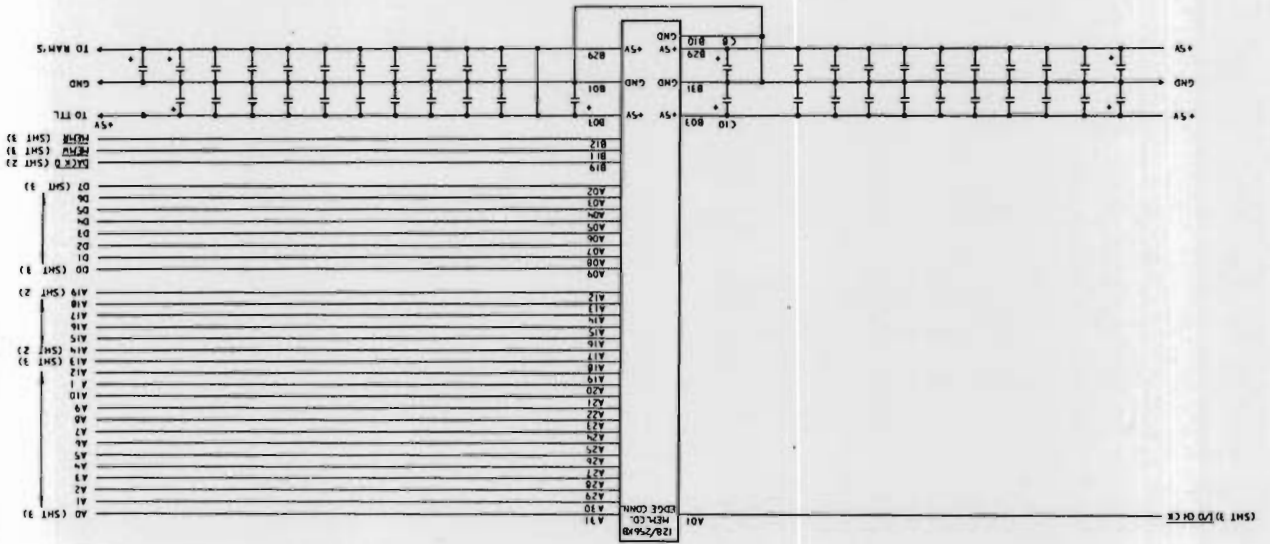
64/256KB MEMORY EXPANSION OPTION

64/256KB Memory Expansion Logic 1 of 4



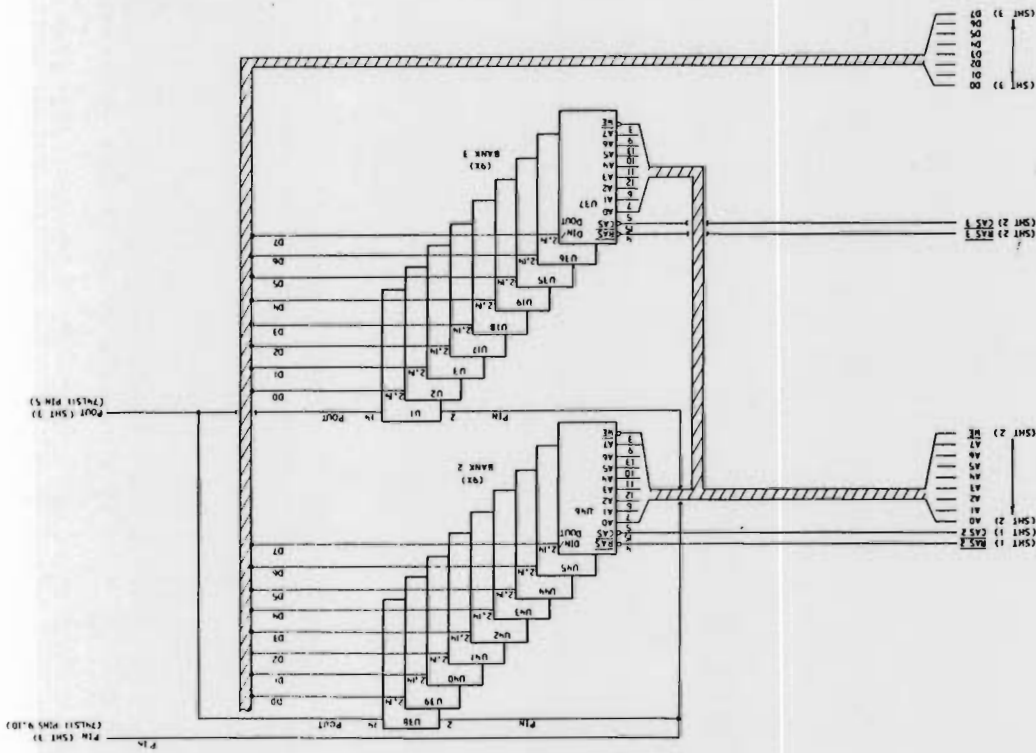
64/256KB MEMORY EXPANSION OPTION

64/256KB MEMORY EXPANSION OPTION

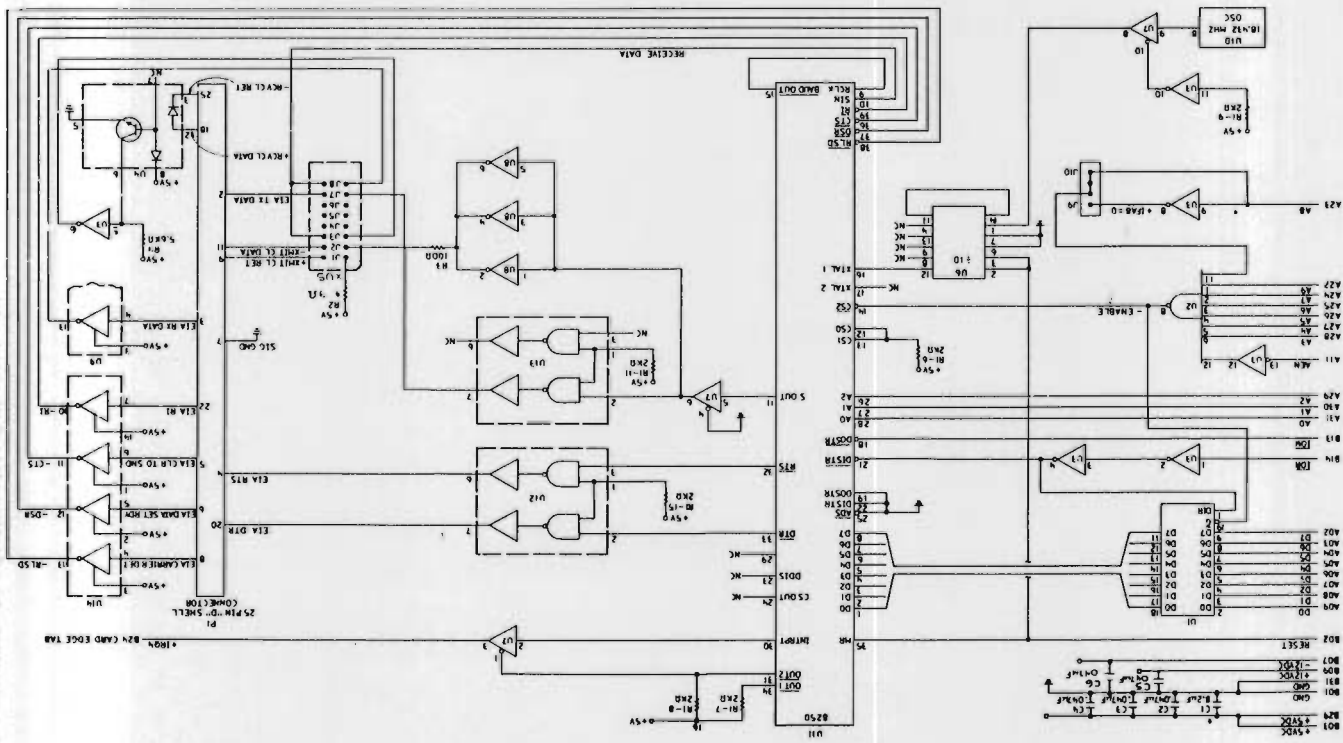


64/256KB Memory Expansion Logic 3 of 4

64/256KB MEMORY EXPANSION OPTION

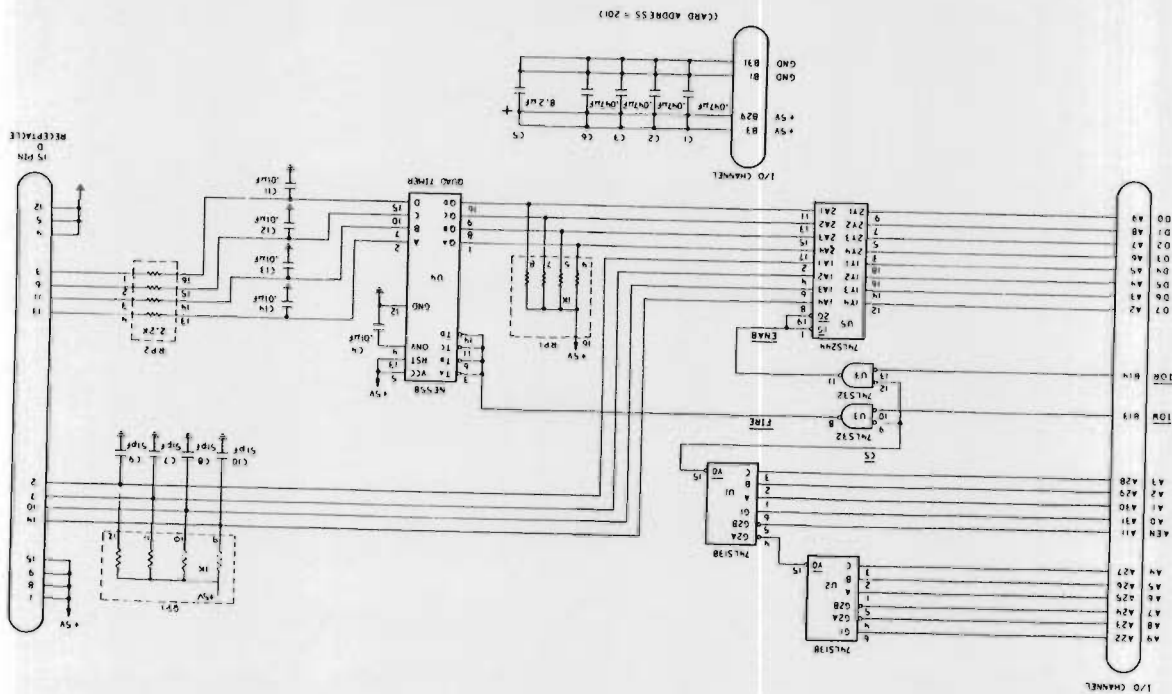


ASYNCHRONOUS COMMUNICATIONS ADAPTER



Asynchronous Communications Adapter

Game Control Adapter



APPENDIX E. UNIT SPECIFICATIONS

System Unit

Size:

Length—19.6" (500 mm)
Depth—16.1" (410 mm)
Height—5.5" (142 mm)

Weight:

Without Diskette Drive Unit—20.9 lbs (9.5 kg)
With Diskette Drive Unit—25 lbs (11.4 kg)

Power Cable:

Length—6'0" (1.83 m)
Size—18 AWG

Environment:

Air Temperature

System ON, 60° to 90° F (15.6° to 32.2° C)
System OFF, 50° to 110° F (10° to 43° C)

Humidity

System ON, 8% to 80%
System OFF, 20% to 80%
Heat Output, 1083 BTU/HR (Maximum)

Noise Levels:

Without Printer, 59 DBS
With Printer, 66 DBS

Electrical:

Nominal—120 VAC
Minimum—104 VAC
Maximum—127 VAC
KVA—3175 maximum
OR
220/240 VAC
180 VAC
259 VAC
KVA—3175 maximum

Keyboard

Size:

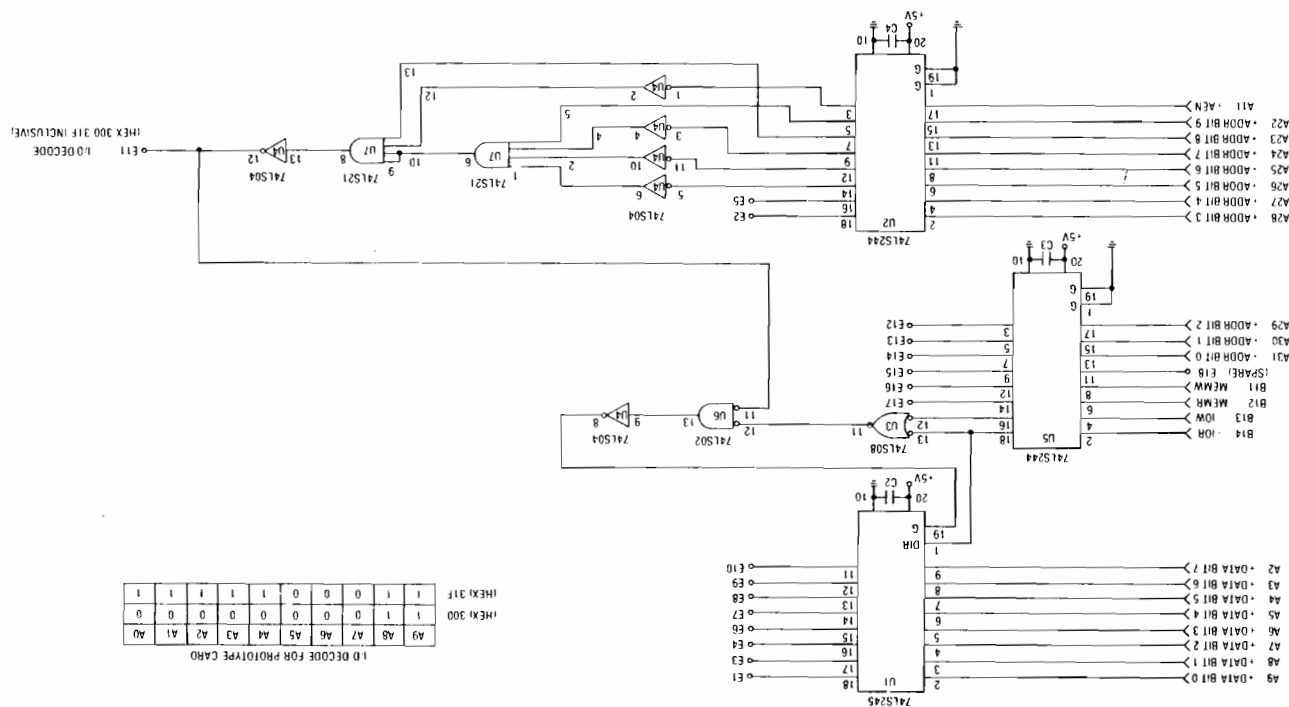
Length—19.6" (500 mm)
Depth—7.87" (200 mm)
Height—2.2" (57 mm)

Weight:

6.5 lbs (14.3 kg)

PROTOTYPE CARD INTERFACE DIAGRAM

Prototype Card Interface Diagram



IBM Monochrome Display

Size:

Length—14.9" (380 mm)

Depth—13.7" (350 mm)

Height—11" (280 mm)

Weight:

17.3 lbs (7.9 kg)

Heat Output:

325 BTU/HR

Power Cable:

Length—3'0" (914 mm)

Size—18 AWG

Signal Cable:

Length—4'0" (1.22 m)

Size—22 AWG

IBM 80 CPS Graphics Printer

Size:

Width—14.7" (374 mm)

Depth—12.0" (305 mm)

Height—4.2" (107 mm)

Weight:

12.0 lbs (5.5 kg)

Power Cable:

Length—6'0" (1.83 m)

Size—18 AWG

Signal Cable:

Length—6'0" (1.83 m)

Size—22 AWG

Heat Output:

341 BTU/HR (Max.)

Electrical (three models available)

| | | | |
|----------|---------|---------|---------|
| Nominal- | 120 VAC | 220 VAC | 240 VAC |
| Minimum- | 104 VAC | 198 VAC | 216 VAC |
| Maximum- | 127 VAC | 242 VAC | 264 VAC |

GLOSSARY

1. **Address Buss:** A set of wires or signals carrying the binary-coded address from the Intel-8088 microprocessor through-out the rest of the IBM Personal Computer System Unit.
2. **AEN:** Address Enable. (Refer to System Board I/O Channel Descriptions).
3. **ALE:** Address Latch Enable. (Refer to System Board I/O Channel Descriptions).
4. **Analog:** (1) Pertaining to representation by means of continuously variable physical quantities. (2) Contrast with digital.
5. **A/N:** Alphanumeric: Pertaining to a character set that contains letters, digits, and usually other characters, such as punctuation marks. Synonymous with alphanumeric.
6. **A0-A19:** Address bits 0-19. (Refer to System Board I/O Channel Descriptions).
7. **APA:** All points addressable graphics.
8. **ASCII:** American Standard Code of Information Interchange. The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), used for information interchange among data processing systems, data communication systems and associated equipment. The ASCII set consists of control characters and graphic characters.
9. **Assembler:** A computer program used to assemble. Synonymous with assembly program.
10. **BASIC:** (Beginner's all-purpose symbolic instruction code). A programming language with a small repertoire of commands and a simple syntax, primarily designed for numerical application.
11. **BAUD:** (1) A unit of signaling speed equal to the number of discrete conditions or signal events per second in Morse code, one bit per second in a train of binary signals, and one 3-bit value per second in a train of signals each of which can assume one of eight different states. (2) In asynchronous transmission, the unit of modulation rate corresponding to one unit of interval per second, i.e. if the duration of the unit interval is 20 milliseconds, the modulation rate is 50 baud.

12. Binary: (1) Pertaining to a selection, choice, or condition that has two possible values or states. (2) Pertaining to a fixed radix numeration system having a radix of two.
13. BIOS: Basic Input/Output System.
14. Bootstrap: A technique or device designed to bring itself into a desired state by means of its own action, e.g. a machine routine whose first few instructions are sufficient to bring the rest of itself into the computer from an input device.
15. Buffer: An area of storage that is temporarily reserved for use in performing an input/output operation, into which data is read or from which data is written. Synonymous with I/O area. A portion of storage for temporarily holding input or output data.
16. Bus: One or more conductors used for transmitting signals or power.
17. Byte: (1) A binary character operated upon as a unit and usually shorter than a computer word. (2) The representation of a character.
18. CLK: Clock. (Refer to System Board I/O Channel Descriptions).
19. Code: (1) A set of unambiguous rules specifying the manner in which data may be represented in a discrete form. Synonymous with coding scheme. (2) A set of items such as abbreviations representing the members of another set. (3) Loosely, one or more computer programs, or part of a computer program. (4) To represent data or a computer program in a symbolic form that can be accepted by a data processor.
20. Computer: A data processor that can perform substantial computation, including numerous arithmetic operations, or logic operations, without intervention by a human operator during the run.
21. CPS: Characters per second.
22. CRC: The cyclic redundancy check character.
23. CRT: (1) A Cathode ray tube display. (2) A display device, such as the IBM Monochrome Display, that uses a cathode ray tube.
24. CTS: Conversational Terminal System. (2) Clear to Send. Associated with modem control.
25. DACK0-DACK3: DMA Acknowledge 0 to 3. (Refer to System Board I/O Channel Description.)
26. Data: (1) A representation of facts, concepts or instructions in a formalized manner suitable for communication, interpretation, or processing by humans or automatic means. (2) Any representations such as characters or analog quantities to which meaning is, or might be assigned.
27. Din Connectors: One of the connectors specified by the Din standardization committee.
28. DIP: "Dual In-Line Package." A widely used container for an integrated circuit. DIP's are pins usually in two parallel rows. These pins are spaced on 1/10" inters and come in different configurations ranging from a 14-pin assembly to a 40-pin configuration.
29. Display: A visual presentation of data.
30. DMA: Direct Memory Access.
31. DO-D7: Data Bits 0 to 7. (Refer to System Board I/O Channel Descriptions.)
32. DRQ1-DRQ3: DMA Request 1 to 3. (Refer to System Board I/O Channel Descriptions.)
33. DSR: Data Set Ready, associated with modem control.
34. DTR: Distribution Tape Reel.
35. Edge Connector: An opening which joins with the end of a circuit board. The purpose of this interface is to send electrical signals back and forth.
36. EIA/CCITT Drives: Electronic Industries Association/Consultative Committee on International Telegraphy and Telephony Drives.
37. EPROM or 'PROM': Term for "Programmable Read-Only Memory." An EPROM or 'PROM' is actually Read-Only Memory (ROM) but the contents may be changed by electrical means. EPROM or 'PROM' information is not destroyed when the power is cut off.
38. Firmware: Memory chips with the software programs already built in.
39. Graphics: Symbols Produced by a process such as handwriting, drawing or printing. Synonymous with graphic symbol.
40. Hexadecimal: Pertaining to a selection, choice, or condition that has sixteen possible values or states. These values or states usually contain 10 digits and six letters A through F. Hexadecimal digits are equivalent to a power of 16.

41. Hertz (Hz): A unit of frequency equal to one cycle per second.
42. High order position: The leftmost position in a string of characters.
43. Input/Output (I/O): Pertaining to a device or to a channel that may be involved in an input process, and, at a different time, in an output process. (2) Pertaining to a device whose parts can be performing an input process and an output process at the same time.
44. Integrated Circuit: A combination of interconnected circuit elements inseparably associated on or within a continuous substrate.
45. Interpreter: A computer program used to interpret. Synonymous with interpretive program.
46. Interrupt: (1) A suspension of a process, such as the execution of a computer program, in such a way that the process can be resumed. (2) To stop a process in such a way that it can be resumed. (3) In data transmission, to take an action at a receiving station that causes the transmitting station to terminate a transmission.
47. I/O Channel: Input/Output Channel. In a data processing system, a functional unit, controlled by the processing unit, that handles the transfer of data between main storage and peripheral equipment.
48. I/O CH CK: I/O-Channel Check. (Refer to System Board I/O Channel Descriptions.)
49. I/O CH RDY: I/O Channel Ready. (Refer to System Board I/O Channel Descriptions.)
50. IMR: Interruption Mask Register.
51. IOR: I/O Read Command. (Refer to System Board I/O Channel Descriptions.)
52. IOW: I/O Write Command. (Refer to System Board I/O Channel Descriptions.)
53. IRQ2-IRQ7: Interrupt Request 2 to 7. (Refer to System Board I/O Channel Descriptions.)
54. K: An abbreviation for the prefix kilo, i.e. 1000 in decimal notation. To the tenth power, 1024 in decimal notation.
55. KB: Kilobyte.
56. Khz: Kilohertz. A unit of frequency equal to 1,000 hertz.

57. Low order position: The rightmost position in a string of characters.
58. Machine Language: (1) A language that is used directly by a machine. (2) Another term for computer instruction code.
59. Memory Address: A two-byte value selecting one specific memory location on a memory map.
60. Memory Location: The most specific part of a memory map that the computer can refer to.
61. Memory Map: The list of memory locations addressed directly by the microprocessor.
62. MEMR: Memory Read Command. (Refer to System Board I/O Channel Descriptions.)
63. MEMW: Memory Write Command. (Refer to System Board I/O Channel Descriptions.)
64. MFM Coded: Modified Frequency Modulation. It is double density encoding of information on a diskette.
65. Mhz: Megahertz. A unit of frequency equal to one million Hertz.
66. Microprocessor: A processing unit, or part of a processing unit, that consists of microcode. In the IBM Personal Computer, the microprocessor is the Intel-8088.
67. Mnemonic: Symbol or symbols used instead of terminology more difficult to remember. Usually a mnemonic has two or three letters.
68. Mode: (1) A method of operation, for example, the binary mode, the interpretive mode, the alphanumeric mode. (2) The most frequent value in the statistical sense.
69. Monitor: (1) A device that observes and verifies the operation of a data processing system and indicates any specific departure from the norm. (2) A television type display such as the IBM Monochrome Display. (3) Software or hardware that observes, supervises, controls, or verifies the operations of a system.
70. Multiplexer: A device capable of interleaving the events of two or more activities or capable of distributing the events of an interleaved sequence to their respective activities.

71. OR: A logic operator having the property that if P is a statement, Q is a statement, R is a statement..., then the OR of P,Q,R, is true if at least one statement is true, false if all statements are false. P OR Q is often represented by P+Q, PVQ. The term is synonymous with boolean add; logic add.
72. "ORed": Past tense of OR.
73. OSC: Oscillator. (Refer to System Board I/O Channel Descriptions.)
74. Output: Pertaining to a device, process, or channel involved in an output process, or to the data or states involved in an output process.
75. Personal Computer: A small home or business computer complete with a System Unit, keyboard, and available with a variety of options such as monochrome display and a dot matrix printer.
76. Pinout: A diagram of functioning pins on a pinboard.
77. Printed Circuit Board: A piece of material, usually fiberglass, which contains a layer of conductive material, usually metal. The metallic layer is then etched and electronic equipment is then attached to the fiberglass. The electronic equipment then has the capacity to transmit electronic signals through the board by way of the etched metal tracks.
78. Program: (1) A series of actions designed to achieve a certain result. (2) To design, write and test computer programs.
79. Read/Write Memory: Random access storage.
80. Reset Drv: Reset Driver. (Refer to System Board I/O Channel Descriptions.)
81. RF Modulator: The device used to convert the composite video signal to the antenna level input of a home TV.
82. ROM: Read-only Memory.
83. ROM BIOS: Read-only Memory/Basic Input Output System.
84. RS 232 Port: Asynchronous Type Communications.
85. RTS: Ready to Send. Associated with modem control.

86. Scan Line: The use of a cathode beam to test the cathode ray tube of a display used with a personal computer.
87. Schematic: The description, usually in diagram form, of the logical structure and physical structure of an entire data base according to a conceptual model.
88. Software: (1) Computer programs, procedures, rules, and possibly associated documentation concerned with the operation of a data processing system. (2) Contrast with hardware.
89. Strobe: (1) An instrument used to determine the exact speed of circular or cyclic movement. (2) A flashing signal displaying an exact event.
90. Text: In ASCII and data communication, a sequence of characters treated as an entity if preceded and terminated by one STX and one ETX transmission control respectively.
91. TX Data: Transmit Data. External connections of the RS 232 Asynchronous Communications Adapter interface.
92. Video: Computer data shown or displayed on a cathode ray tube monitor or display.

BIBLIOGRAPHY

IBM Publications

1. IBM Personal Computer Guide to Operations
General information on using the IBM Personal Computer.
2. IBM Personal Computer Hardware and Service
Information on hardware and steps necessary when servicing the IBM Personal Computer.
3. IBM Personal Computer BASIC
Information for programmers who are using BASIC.
4. IBM Personal Computer Disk Operating System (DOS)
Information for programmers who are using DOS.
5. IBM Personal Computer MACRO Assembler
Information for experienced assembly language programmers using the Macro Assembler.
6. IBM Personal Computer Pascal Compiler
Information for programmers who are familiar with the Pascal language.

Other Related Publications

7. NATIONAL SEMICONDUCTOR
INS 8250 Asynchronous Communications Element
This book documents physical and operating characteristics of the INS 8250.
8. INTEL
The 8086 Family Users Manual
This manual introduces the 8086 family of microcomputing components and serves as a reference in system design and implementation.
9. INTEL
8086/8087/8088 Macro
ASSEMBLY Language
Reference Manual for 8088/8085 Based Development System
The manual describes the 8086/8087/8088 Macro Assembly Language, and is intended for use by persons who are familiar with assembly language.
10. MOTOROLA
The complete Microcomputer Data Library
This book can provide additional information on the Motorola 6845 CRT Controller used in the IBM Monochrome Display and Parallel Printer Adapter, and the Color/Graphics Monitor Adapter.

INDEX

A

- A0-A19 (Address Bits 0-19) 2-10
- Accessible Registers 2-152
- A.C. Output 2-44
- Adapters
 - Asynchronous Communications (RS232) 2-141
 - Color/Graphics 2-61
 - 5 1/4" Diskette Drive 2-107
 - Game Control 2-135
 - IBM Monochrome Display and Parallel Printer 2-47
 - Parallel Printer 2-75
- Adapter Attribute Relationship 2-61
- Adapter Inputs 2-125
- Adapter Outputs 2-124
- Address Bits (A0-A19) 2-10
- Address Decode 2-136
- Address Enable (AEN) 2-12
- Address Latch Enable (ALE) 2-10
- Address Strobe (ADS) 2-147
- ADS (Address Strobe) 2-147
- AEN (Address Enable) 2-12
- ALE (Address Latch Enable) 2-10
- Algorithms 3-8
- All Points Addressable (APA) 2-55
- Alphanumeric Mode 2-59
- American Standard Code for Information Interchange (ASCII) 1-2
- Analog Input 2-140
- A/N (Alphanumeric) 2-59
- Appendices
 - A - ROM BIOS Listing A-1
 - B - Assembly Instruction Set Reference B-1
 - C - Of Characters, Keystrokes and Color C-1
 - D - Logic Diagrams D-1
 - E - Unit Specifications E-1
- APA (All Points Addressable) 2-55
- ASCII - (See American Standard Code for Information Interchange) 1-2

ASCII (Extended) 3-11
 Assembly Instruction Set Reference Appendix B
 Assembly Language Reference Guide (See Bibliography)
 Asynchronous Communications Adapter
 Block Diagram 2-142
 Current Loop Interface 2-145
 Input/Output Decode 2-143
 Input/Output Signals 2-151
 Input Signals 2-147
 Interface Description 2-144
 Interface Specifications 2-165
 Interrupt 2-144
 Interrupt Enable Register 2-159
 Interrupt Identification Register 2-157
 INS 8250 Accessible Registers 2-151
 INS 8250 Functional Pin Description 2-147
 INS 8250 Line Control Register 2-152
 INS 8250 Programmable Baud Rate Generator 2-153
 Line Status Register 2-151
 Logic Diagram D-54
 Modem Control Register 2-160
 Modem Status Register 2-161
 Modes of Operation 2-143
 Output Signals 2-150
 Programming Considerations 2-151
 Receiver Buffer Register 2-162
 Reset Functions 2-151
 Selecting the Interface Format 2-164
 Transmitter Holding Register 2-163

B

BASIC (Beginner's all-purpose symbolic instruction code)
 80 Interpreter 1-1
 Reserved Interrupts 3-23
 Screen Editor Special Functions 3-19
 Workspace Variables 3-23
 Baud (See INS 8250 Programmable Baud Rate Generator) 2-150
 Baud Out (BAUDOUT) 2-150
 BEL (Bell) 2-100
 Bell (BEL) 2-100
 Berg Pin Connectors 2-73

BIOS
 (Basic Input Output System) 3-2
 Cassette Logic 3-8
 Memory Map 3-7
 Interrupt Vector Listing 3-3
 Parameter Passing 3-2
 Programming Tip 3-2
 ROM (Read Only Memory) 3-2
 Use of 3-2
 Vectors With Special Meaning 3-5
 BIT
 I/O Address 2-52
 I/O Bit Map 2-29
 Output Port Cassette 2-25
 Output Port Speaker 2-28
 Status Register 2-69
 Block Diagrams
 Asynchronous Communications Adapter 2-142
 Cassette Motor Control 2-26
 Cassette Interface Read 2-25
 Cassette Interface Write 2-26
 Color/Graphics Monitor Adapter 2-57
 5 1/4" Diskette Drive Adapter 2-108
 Game Control Adapter 2-135
 Keyboard Interface 2-15
 IBM Monochrome Display Adapter 2-48
 Parallel Printer Adapter 2-76
 System 1-4
 Bootstrap 3-3
 Byte:
 Attribute Definition 2-60
 Display Buffer 2-71

C

CAN (Cancel) 2-93
 Cancel (CAN) 2-93
 Carriage Return (CR) 2-92
 Cassette
 Circuit Block Diagrams 2-25
 Data Record Architecture 3-10
 Error Recovery 3-10
 Interface Connector Specifications 2-27
 Interrupt 15 3-8

- Cassette (continued)
 - Jumpers 2-25
 - Logic (BIOS) 3-8
 - Read 3-9
 - Write 3-8
- Character
 - Codes 3-11
 - Generator 2-58
- Of Characters, Keystrokes and Color Appendix C
- Set (00-7F) Quick Reference C-12
- Set (80-FF) C-13
- Chip Select Out (CSOUT) 2-150
- Clear To Send (CTS) 2-148
- Clock (4.77 MHz) 2-10
- CLK (Clock) 2-10
- CNTRL (Control) 3-14
- Color/Graphics Monitor Adapter
 - Block Diagram 2-57
- Character Generator 2-58
- Color/Graphics Mode 2-61
- Color Select Register 2-67
- Composite Color Generator 2-58
- Description of Basic Operations 2-63
- Display Buffer 2-58
- Graphics Storage Map 2-62
- I/O Address and Bit Map 2-71
- Interrupt Level 2-70
- Logic Diagrams D-27
- Major Component Definitions 2-58
- Memory Requirements 2-70
- Mode Register Summary 2-69
- Mode Select Register 2-68
- Mode Set and Status Registers 2-69
- Modes of Operation 2-59
- Monitor Adapter Auxiliary Connectors 2-73
- Monitor Adapter Direct Drive and Composite Interface Pin Assignment 2-72
- Monochrome Display Adapter Vs. Color/Graphics Adapter Attribute Relationship 2-61
- Motorola 6845 CRT Controller 2-58
- Programming the 6845 Controller 2-65
- Programming the Mode Control and Status Registers 2-67
- 6845 Register Description 2-66
- Sequence of Events 2-70
- Timing Generator 2-58

- Composite Phone Jack 2-72
- Computer (IBM Personal) 1-1
- Command Phase 2-111
- Command Status Registers
 - Register 0 2-118
 - Register 1 2-119
 - Register 2 2-120
 - Register 3 2-121
- Control Codes
 - CPS (Characters Per Second) 2-80
 - CPU (Central Processing Unit) See System Board, Microprocessor 2-3
 - CR (Carriage Return) 2-92
- CRT:
 - (Cathode Ray Tube) 2-47
 - Output Port 1 (I/O Address '3B8') 2-52
 - Status Port (I/O Address '3BA') 2-53
- CTS (Clear To Send) 2-148
- CSOUT (Chip Select Out) 2-150
- Current 2-44
- Cursor 2-14

D

- D0-D7 (Data Bits 0-7) 2-10
- DACK0 - DACK3 (DMA Acknowledge 0 to 3) 2-12
- DATA
 - Bit (D0-D7) 2-10
 - Bus Buffer/Driver 2-136
 - Input Strobe (DISTR, DISTR) 2-147
 - Output Strobe (DOSTR, DOSTR) 2-147
 - Rates 2-43
- Record Architecture 3-10
- Set Ready (DSR) 2-148
- Data Flow (System) 2-6
- DC 2 (Device Control 2) 2-93
- DC 4 (Device Control 4) 2-93
- DC Output 2-44
- DEL (Delete) 2-85
- Delete (DEL) 2-85
- Device Control 2 (DC 2) 2-93

Device Control 4 (DC 4) 2-93
 Digital Output Register (DOR) 2-109
 DIN (Connectors) 2-5
 DIP (Dual In-Line Package) 2-28
 Diskettes 2-129
 Diskette Drive (5 1/4") 2-128
 Diskette Drive (5 1/4") Adapter
 Adapter Inputs 2-124
 Adapter Outputs 2-125
 Block Diagram 2-108
 Command Status Registers 2-118
 Command Summary 2-114
 Comments (Programming) 2-122
 Digital Output Register 2-109
 Drive A and B Interface 2-124
 Drive Constants 2-122
 DPC Registers 2-123
 External Interface Specifications 2-127
 Floppy Disk Controller 2-109
 Functional Description 2-109
 Internal Interface Specifications 2-126
 Logic Diagrams D-37
 Programming Considerations 2-112
 Programming Summary 2-121
 System I/O Channel Interface 2-122
 Display (See IBM Monochrome Display) 2-53
 Display Buffer 2-58
 DISTR, DISTR (Data Input Strobe) 2-147
 Divisor Latch 2-154
 DMA (Direct Memory Access) 2-4
 DOS Special Functions 3-19
 DOSTR, DOSTR (Data Output Strobe) 2-147
 Drive Constants 2-122
 Drive Disable 2-150
 DRQ1 – DRQ3 (DMA Request 1 to 3) 2-12
 DSR (Data Set Ready) 2-149
 DTR (Data Terminal Ready) 2-150
 Dual In-Line Package Switches (DIP) 2-34

F

Edge Connector 2-126
 Encoding 3-11
 Error Recovery 3-10
 ESC (Escape) 2-93

ESC A (Escape A) 2-95
 ESC C (Escape C) 2-96
 ESC D (Escape D) 2-96
 ESC E (Escape E) 2-97
 ESC F (Escape F) 2-97
 ESC G (Escape G) 2-97
 ESC H (Escape H) 2-97
 ESC J (Escape J) 2-97
 ESC K (Escape K) 2-98
 ESC L (Escape L) 2-100
 ESC N (Escape N) 2-100
 ESC O (Escape O) 2-100
 ESC S (Escape S) 2-100
 ESC T (Escape T) 2-101
 ESC U (Escape U) 2-101
 ESC W (Escape W) 2-101
 ESC Y (Escape Y) 2-101
 ESC Z (Escape Z) 2-101
 ESC 0 (Escape 0) 2-94
 ESC 1 (Escape 1) 2-94
 ESC 2 (Escape 2) 2-94
 ESC 3 (Escape 3) 2-94
 ESC 6 (Escape 6) 2-94
 ESC 7 (Escape 7) 2-94
 ESC 8 (Escape 8) 2-95
 ESC 9 (Escape 9) 2-95
 ESC – (Escape Minus) 2-93
 ESC < (Escape Less Than) 2-95
 Execution Phase 2-111
 Extended Codes 3-13

F

FDC (Floppy Disk Controller) 2-109
 Floppy Disk Controller (FDC) 2-109
 Fonts 2-58
 Functions and Conditions of DIP Switch 1 2-83
 Functions and Conditions of DIP Switch 2 2-84
 Functional Description
 5 1/4" Diskette Drive Adapter 2-109
 Game Control Adapter 2-136
 INS 8250 2-147

G

- Game Control Adapter
 - Address Decode 2-136
 - Block Diagram 2-135
 - Connector Specifications 2-140
 - Data Bus Buffer/Driver 2-136
 - Functional Description 2-136
 - Interface Description 2-137
 - Joystick Positions 2-136
 - Joystick Schematic 2-139
 - Trigger Buttons 2-136
- Glossary G-1
- GND (Ground) 2-12
- Graphics Character Extensions (Interrupt 1 FH) 3-6
- Graphics Mode (Color) 2-61
- Graphics Printer (IBM 80 CPS) 2-80
- Graphics Storage Map 2-62

H

Hardware

- Asynchronous Communications Adapter 2-141
- Color/Graphics Monitor Adapter 2-55
- 5 1/4" Diskette Drive 2-128
- 5 1/4" Diskette Drive Adapter 2-107
- Game Control Adapter 2-135
- IBM Monochrome Display 2-53
- IBM Monochrome Display and Parallel Printer Adapter 2-47
- Memory Expansion Options 32KB and 64KB 2-131
- Printer 2-80
- Parallel Printer Adapter 2-75
- Power Supply 2-43
- System Board 2-3

Hardware

- Overview 1-1
- Data Flow 2-6
- Hertz (Hz) 1-2
- Horizontal Drive 2-53
- Horizontal Tab (HT) 2-91
- HT (Horizontal Tab) 2-91

I

- IBM 80 CPS Graphics Printer 2-80
- IBM Monochrome Display 2-53
- IBM Monochrome Display and Parallel Printer Adapter
 - Block Diagram 2-47
 - Data Rates 2-49
 - Direct Drive Interface and Pin Assignment 2-54
 - DMA Channel 2-52
 - Interrupt and DMA Response Requirements 2-49
 - Interrupt Levels 2-52
 - I/O Address and Bit Map 2-52
 - Lines Used 2-49
 - Loads 2-49
 - Memory Requirements 2-51
 - Modes of Operation 2-50
 - Programming the 6845 CRT Controller 2-51
 - Sequence of Events 2-51
- Important Operating Characteristics 2-46
- IER (Interrupt Enable Register) 2-159
- IIR (Interrupt Identification Register) 2-157
- Input/Output Signals:
 - Data (D0, D7) 2-151
 - External Clock Input/Output (XTAL1, XTAL2) 2-151
 - Input Requirements (Power Supply) 2-44
- Input Signals
 - Address Strobe (ADS) 2-147
 - Chip Select (CS0, CS1, CS2) 2-147
 - Clear To Send (CTS) 2-148
 - Data Input Strobe (DISTR, DISTR) 2-147
 - Data Output Strobe (DOSTR, DOSTR) 2-147
 - Data Set Ready (DSR) 2-149
 - Master Reset (MR) 2-148
 - Received Line Signal (RCLK) 2-148
 - Receiver Clock (RCLK) 2-148
 - Register Select (A0, A1, A2) 2-148
 - Ring Indicator (RI) 2-149
 - Serial Input (SIN) 2-148

- Interface Diagram
 - Asynchronous Communications Adapter 2-165
 - Cassette Connector Specifications 2-27
 - Color/Graphics Monitor Adapter 2-72
 - 5 1/4" Diskette Drive Adapter External 2-127
 - 5 1/4" Diskette Drive Adapter Internal 2-126
 - Game Control 2-140
 - IBM Monochrome Display Direct Drive 2-54
 - Keyboard Connector Specifications 2-24
 - Parallel Printer 2-79
 - INTRPT (Interrupt) 2-150
 - Interrupt Control Functions 2-158
 - Intel 8048 (Keyboard Microcomputer) 2-14
 - Intel 8088 (System Unit Microprocessor) 2-3
 - I/O (Input/Output)
 - Address Map 2-71
 - Channel 2-8
 - Channel Description (System Board) 2-10
 - Diagram 2-9
 - I/O CH CK (I/O Channel Check) 2-10
 - I/O CH RDY (I/O Channel Ready) 2-11
 - IOR (I/O Read Command) 2-11
 - IOW (I/O Write Command) 2-11
- Interrupts
 - And DMA Response Requirements 2-49
 - Enable Register 2-159
 - Identification Register 2-157
 - Levels (0-XX) 2-52
 - Vector Listing 3-3
 - Vectors (0-7F) 3-21
 - 1 CH - Timer Tick 3-5
 - 1 DH - Video Parameters 3-5
 - 1 EH - Diskette Parameters 3-5
 - 1 FH - Graphics Character Extensions 3-6
 - 15 3-8

K

- KB - Kilobyte (See Memory Expansion Options)
- Keyboard
 - Break 3-16
 - Character Codes 3-11
 - Diagram 2-16
 - Encoding 3-11

- Keyboard (continued)
 - Extended Codes 3-13
 - Extended Functions 3-13
 - Interface Block Diagram 2-15
 - Interface Connector Specifications 2-24
 - Pause 3-16
 - Print Screen 3-16
 - Scan Codes 2-23
 - Shift States 3-14
 - Shift Key Priorities 3-15
 - Special Handling 3-15
 - System Reset 3-15
 - Usage 3-17
- Kilobyte (KB) (See Memory Expansion Options)

L

- LF (Line Feed) 2-91
- Light Pen
 - Interface 2-73
 - Mode Control and Status Register 2-67
 - Register Description 2-66
- Line Control Register (LCR) 2-138
- Line Feed 2-91
- Line Status Register (LSR) 2-141
- Lines Used 2-49
- Loads 2-49
- Logic Diagrams Appendix D
- Low Memory Maps
 - BASIC and DOS Reserved Interrupts (80-3FF) 3-22
 - BASIC Workspace Variables 3-23
 - Interrupt Vectors (0-7F) 3-21
 - Reserved Memory Locations (400-5FF) 3-22

M

- Major Component Definitions 2-58
- Matrix Printer (See Graphics Printer) 2-80
- Megabyte 2-3
- Memory
 - BIOS Map 3-7
 - Map (System) 2-31
 - Module Description 2-132
 - Module Pin Configuration 2-132

Memory (continued)

- Other Read/Write Usage 3-6
- Requirements (Color/Graphics) 2-70
- Requirements (IBM Monochrome) 2-51
- System Board Switch Settings 2-36
- 32KB, 64KB and 64/256KB Memory Expansion Option Switch Settings 2-38
- Memory Expansion Options
 - Memory Module Description 2-14
 - Memory Module Pin Configuration 2-132
 - Operating Characteristics 2-131
 - Switch Configurable Starting Address 2-133
- MEMR (Memory Read Command) 2-11
- MEMW (Memory Write Command) 2-11
- MFH (Modified Frequency Modulation) 2-114
- MHz (Megahertz) 2-53
- Microprocessor 2-3
- Microsecond 2-3
- Mnemonic B-18
- Mode Set and Status Register 2-58
- Modem Control Register 2-160
- Modem Status Register 2-161
- Modes of Operation
 - Asynchronous Communications Adapter 2-143
 - Color/Graphics Monitor Adapter 2-59
 - IBM Monochrome Display Adapter 2-50
 - Mode Register Summary 2-69
 - Mode Select Register 2-68
 - Monitor Type Switch Settings 2-35
 - Monochrome Display (IBM) 2-53
 - Monochrome Display and Parallel Printer Adapter (See IBM)
 - Motorola 6845 CRT Controller 2-58
 - MR (Master Reset) 2-148

N

- NMI (Non-Maskable Interrupt)
 - of the 8088 2-4
 - to the 8088 2-8
- NMI Mask. Reg - (I/O Address Map) 2-29
- Nominal Power Requirements 2-30
- NUL (Null) 2-91
- Null (NUL) 2-91
- NUM LOCK 3-14

O

- Of Characters, Keystrokes and Color C-1
- OHM Resistors 2-77
- Operating Characteristics
 - Memory Expansion Options 2-131
 - Monochrome Display 2-53
 - Power Supply 2-46
- Options
 - Asynchronous Communications Adapter 2-12
 - Color/Graphics Monitor Adapter 2-55
 - 5 1/4" Diskette Drive 2-128
 - 5 1/4" Diskette Drive Adapter 2-107
 - Game Control Adapter 2-135
 - IBM 80 CPS Graphics Printer 2-80
 - IBM Monochrome Display 2-53
 - Memory Expansion Options 2-131
 - Parallel Printer Adapter 2-75
 - Prototype Card 2-167
 - "ORed" 2-75
 - OSC (Oscillator) 2-10
 - Other Read/Write Memory Usage 3-6
 - OUT PORT 2-49
 - Output
 - AC 2-44
 - Address 2-77
 - DC 2-44
 - Port 2-75
 - OUT 1 2-150
 - OUT 2 2-150
 - Output Signals
 - Baud Out (BAUDOUT) 2-150
 - Chip Select Out (CSOUT) 2-150
 - Data Terminal Ready (DTR) 2-150
 - Driver Disable (DDIS) 2-150
 - Interrupt (INTRPT) 2-150
 - Output 1 (OUT1) 2-150
 - Output 2 (OUT2) 2-150
 - Request to Send (RTS) 2-150
 - Serial Output (SOUT) 2-150
 - Overview (Hardware) 1-1
 - Over Voltage/Current Protection 2-46

P

- Parallel Printer Adapter
 - Block Diagram 2-76
 - Character Sets 2-102
 - Description 2-47
 - DMA Channel 2-52
 - IBM 80 CPS Graphics Printer 2-80
 - I/O Address and Bit Map 2-52
 - Interrupt Levels 2-52
 - Logic Diagram D-33
 - Parallel Interface Description 2-84
 - Printer Control Codes 2-91
 - Printer Modes 2-90
 - Printer Specifications 2-81
 - Programming Considerations 2-77
 - Programming the 6845 CRT Controller 2-51
 - Sequence of Events 2-51
 - Setting the DIP Switches 2-83
 - Timing 2-89
- Parameters, 6845 Initialization 2-51
- Parameter Passing, ROM BIOS 3-2
- Parity Flag, 8080 Flags B-1
- Pause, BIOS Cassette Logic Special Handling 3-16
- Pin Connectors
 - P2-6 Pin Berg Strip for Light Pen Connector 2-73
 - P1-4 Pin Berg Strip for RF Modulator 2-73
 - 9 Pin Connector, Color Direct Drive 2-71
 - 9 Pin Connector, IBM Monochrome Display 2-53
 - 25 Pin Connector, Parallel Printer Adapter Block Diagram 2-76
 - 25 Pin 'D' Shell Connector, Asynchronous Adapter Block Diagram 2-142
 - 15 Pin 'D' Shell Connector, Game Controller Adapter (Analog Input Connector Specifications) 2-140
 - 25 Pin 'D' Shell Connector, Parallel Printer Adapter 2-79
 - 5 Pin Din Connector, Keyboard Interface Connector Specifications 2-24
 - 15 Pin Male 'D' Shell Connector, Joystick Schematic 2-139
- Planar (See System Board 2-3; Intel 8088 2-3)
- Power-on Self Test
 - System Board 2-4
 - Keyboard 2-14
- Power Supply 2-43
 - AC Output 2-44
 - DC Output 2-44
 - Important Operating Characteristics 2-46
 - Over Voltage/Current Protection 2-46
 - Signal Requirements 2-46
 - Input Requirements 2-44
 - Power Supply Connectors and Pin Assignments 2-45
 - Power Supply Location 2-44
- Preface i
- Print Screen, Special Handling 3-16
- Printer, IBM 80 CPS Graphics 2-80
- Printer Specifications 2-81
- Programmable Peripheral Interface (PPI) 8255 A-5 2-25
- Programming Considerations
 - Asynchronous Communications Adapter
 - Asynchronous Communications Reset Functions 2-151
 - INS 8250 Accessible Registers 2-152
 - INS 8250 Line Control Register 2-152
 - INS 8250 Programmable Baud Rate Generator 2-153
 - Interrupt Identification Register 2-157
 - Interrupt Enable Register 2-159
 - Line Status Register 2-155
 - Modem Control Register 2-160
 - Modem Status Register 2-161
 - Receiver Buffer Register 2-162
 - Transmitter Holding Register 2-163
 - Color/Graphics Monitor Adapter
 - Color Select Register 2-67
 - Control and Status Register 2-67
 - I/O Address and Bit Map 2-71
 - Interrupt Level 2-70
 - Mode Register Summary 2-69
 - Mode Select Register 2-68
 - Programming the 6845 Controller 2-65
 - Programming the Modem Control and Status Register 2-67
 - 6845 Register Descriptions 2-66
 - Status Register 2-69
 - Sequence of Events 2-70
- 5 1/4" Diskette Drive Adapter
 - Command Status Registers 2-118
 - Command Summary 2-114
 - Symbol Descriptions 2-112

Programming Considerations (continued)
 IBM Monochrome Display and Parallel Printer Adapter
 DMA Channel 2-52
 I/O Address and Bit Map 2-52
 Interrupt Levels 2-52
 Memory Requirements 2-51
 Sequence of Events 2-51
 Programming the CRT Controller 2-51
 Programming the 6845 CRT Controller 2-51
 Comments 2-122
 DPC Registers 2-121
 Drive Constants 2-122
 Prototype Card 2-167
 I/O Channel Interface 2-169
 Prototype Card to System Board Interface 2-168
 System Loading and Power Limitations 2-172
 Prototype Card External Interface 2-173

R

RAS 2-132
 Rating Amps, Over Voltage/Current Protection 2-46
 Ready Line 2-8
 Read Block 3-9
 Read Data 2-125, 2-126 (5 1/4" Diskette Drives, External) 2-127
 Read Status (Parallel Printer Adapter Block Diagram) 2-76
 Read/Write Memory
 Color Monitor 2-60
 Color TV 2-59
 Future Expansion in I/O Channel, 384KB 2-31
 Graphics Storage Map 2-62
 Hardware Overview 1-1
 I/O Address Map 2-29
 Memory Address Space 2-71
 Memory Expansion Options 2-131
 User 3-7
 Receive Circuit 2-145
 Receiver Clock, (RCLK) 2-146
 Recording Medium 2-129
 Refresh Cycles 2-8

Registers, Address
 Command Status 2-118
 Data 2-9
 Initialization
 Parameters 2-51
 INS 8250 Accessible 2-152
 INS 8250 Line Control 2-152
 Interrupt Enable 2-159
 Line Control 2-152
 Line Status 2-155
 Main Status 2-9
 Modem Control 2-160
 Modem Status 2-161
 Receiver Buffer 2-162
 Transmitter Holding 2-163
 6845 2-65
 Register File
 Color Select 2-67
 6845 Data 2-51
 Description 2-66
 6845 Index 2-51
 6845 Initialization Parameters 2-51
 Mode Select 2-69
 Status 2-67, 2-69
 Requirements
 Input (Power Supply) 2-44
 Memory 2-70
 Signal (Power Supply) 2-46
 RESET DRV (Reset Drive) 2-10
 I/O Channel Description 2-137
 ROM Address Space, 256KB 2-31
 =RESET 2-123
 Reserved Memory Locations (400-5FF) 3-22
 Result Phase 2-111
 Response Requirements, Interrupt and DMA 2-49
 Reverse Video
 Modes of Operation 2-50
 Color Graphics Monitor Adapter 2-55
 RF Modulator
 Auxiliary Video Connector P1-4 Pin Berg Strip 2-73
 Color Graphics Monitor Adapter 2-55
 Interface 2-73
 Ring Indicator 2-145

ROM

- Character Generator 2-58
- Color Graphics Mode 2-62
- Color Graphics Monitor Adapter 2-56
- Diagram 2-13
- Hardware Overview 1-1
- Keyboard 2-14
- Memory Expansion Options 2-131
- System Board 2-3, 2-4
- System Board Component Diagram 2-13
- System Memory Map 2-33
- ROM and System Usage 3-1
- ROM BIOS
 - BIOS Cassette Logic 3-8
 - BIOS Memory Map 3-8
 - Cassette Read 3-9
 - Cassette Write 3-9
 - Data Record Architecture 3-10
 - Description 3-2
 - Error Recovery 3-10
 - Interrupt 15 3-8
 - Interrupt ICH Timer Tick 3-5
 - Interrupt IDH Video Parameters 3-5
 - Interrupt IEH Diskette Parameters 3-5
 - Interrupt IFH Graphics Character Extensions 3-6
 - Interrupt Vector Listing 3-3
 - Other Read/Write Memory Usage 3-7
- ROM BIOS Listing Appendix A
- ROS (Read Only Storage) (See ROM)
- RQM, Request for Master 2-110
- RS232-C (See Asynchronous Communications Adapter) 2-141
- RTS (Ready To Send) 2-141
- R/W (Read/Write)
 - Symbol Description 2-99

S

- Scan Codes 2-23
- Screen 2-53
- Schematic (See Logic Diagrams)
- SCROLL LOCK 3-15
- Selecting the Interface Format 2-164
- Sequence of Events 2-70
- Serial Input (SIN) 2-148
- Serial Output (SOUT) 2-150
- Setting the DIP Switches 2-83
- Shift In 2-93
- Shift Out 2-92
- Shift States 3-14
- SI (Shift In) 2-93
- Signal Requirements 2-46
- SIN (Serial Input) 2-148
- SO (Shift Out) 2-92
- Software Algorithms 3-8
- SOUT (Serial Output) 3-136
- Speaker
 - Drive System Block Diagram 2-28
 - Interface 2-28
 - Special Handling 3-15
 - Special Timing 2-49
 - Specifications
 - 5 1/4" Diskette Drive 2-130
 - Printer 2-81
 - System Appendix E
- Status Registers
 - Color/Graphics 2-69
 - 0 (5 1/4" Diskette Drive Adapter) 2-118
 - 1 (5 1/4" Diskette Drive Adapter) 2-119
 - 2 (5 1/4" Diskette Drive Adapter) 2-120
 - 3 (5 1/4" Diskette Drive Adapter) 2-121
 - Storage (See Memory)
 - Strobe
 - Address 2-147
 - Data Output 2-147
 - Summary of Available Colors 2-65
 - Switch Settings
 - Configurable Start Address 2-133
 - 5 1/4" Diskette Drives 2-35
 - Memory Expansion Option 2-38
 - Monitor Type 2-35
 - System Board Memory 2-36
- System Board
 - Cassette Interface Connector Specifications 2-27
 - Cassette Jumpers 2-25
 - Cassette User Interface 2-25
 - Circuit Block Diagrams (Cassette) 2-25
 - Component Diagram 2-13
 - Data Flow 2-6
 - 5 1/4" Diskette Drive Switch Settings 2-35
 - I/O Address Map 2-29

System Board (continued)

- I/O Channel 2-8
- I/O Channel Description 2-10
- I/O Channel Diagram 2-9
- Keyboard 2-14
- Keyboard Diagram 2-16
- Keyboard Interface Block Diagram 2-15
- Keyboard Interface Connector Specifications 2-24
- Keyboard Scan Codes 2-23
- 32KB, 64KB and 64/256KB Memory Expansion Option Switch Settings 2-38
- Memory Switch Settings 2-36
- Monitor Type Switch Settings 2-35
- Speaker Drive System Block Diagram 2-28
- Speaker Interface 2-28
- System Memory Map 2-31
- System Expansion Slots (See I/O Slots) 2-3
- System Memory Map 2-31
- System Memory Map (16KB Increments) 2-32
- System Unit 1-1
- System Unit Power Connector 2-45
- System Usage (ROM and) 3-1

T

- Tab (Vertical) 2-92
- T/C (Terminal Count) 2-12
- Terminal Count (T/C) 2-12
- THR (Transmitter Holding Register) 2-163
- Timing Generator 2-58
- Transmit Circuit 2-145
- Transmit Data (TX) 2-145
- Transmitter Holding Register (THR) 2-163
- Transmitter Output and Receiver Input 2-144
- Trigger Buttons 2-136
- TX Data (Transmit Data) 2-145

U

- Unit Specifications Appendix E
- Usage (Keyboard) 3-17
- Use of BIOS 3-2
- User Interface (Cassette) 2-25

V

- Vectors
 - (0-7F Interrupt) 3-21
 - (Interrupt Listing) 3-3
 - Vectors With Special Meaning
 - Interrupt 1CH Timer Tick 3-5
 - Interrupt 1DH Video Parameters 3-5
 - Interrupt 1EH Diskette Parameters 3-5
 - Interrupt 1FH Graphics Character Extensions 3-6
 - Other Read/Write Memory Usage 3-6
- Vertical Drive 2-53
- Video
 - Monitor 2-72
 - (Reverse) 2-55
 - Signal 2-53
- Voltage Interchange Information 2-146
- Voltage
 - Power Supply 2-43
 - I/O Channel 2-8
 - System Board I/O Channel Description 2-10
 - Keyboard Interface Connector Specifications 2-24
 - Cassette Interface Connector Specifications 2-27
 - Speaker Interface 2-28
 - Power Supply 2-43
 - Power Supply Location 2-44
 - Important Operating Characteristics 2-46
- Color/Graphics Monitor Adapter Direct Drive and Composite Interface Pin Assignment 2-72
- Color/Graphics Monitor Adapter Auxiliary Video Connectors 2-73
- Printer Specifications 2-81
- Game Controller Adapter (Analog Input) Connector Specifications 2-140
- Voltage Interchange Information, Asynchronous Communications Adapter 2-146
- Selecting the Interface Format, Asynchronous Communications Adapter 2-164

W

Workspace (BASIC Variables) 3-23
Write (Cassette) 3-8
Write (Cassette Interface Hardware) 2-26
Write Data 2-114
Write Enable 2-114
Write Protect 2-115

Numerics

5 1/4" Diskette Drive 2-128
5 1/4" Diskette Drive Adapter 2-107
32, 64 and 64/256KB Memory Expansion Options 1-3
80 CPS Graphics Printer, IBM 2-80
80 Interpreter, BASIC 1-1
6845 CRT Controller 2-58
8080 Parity Flags B-1
8088, Intel 2-3
8250 INS Accessible Registers 2-152
RS232C-A Asynchronous Communications Adapter 2-141